

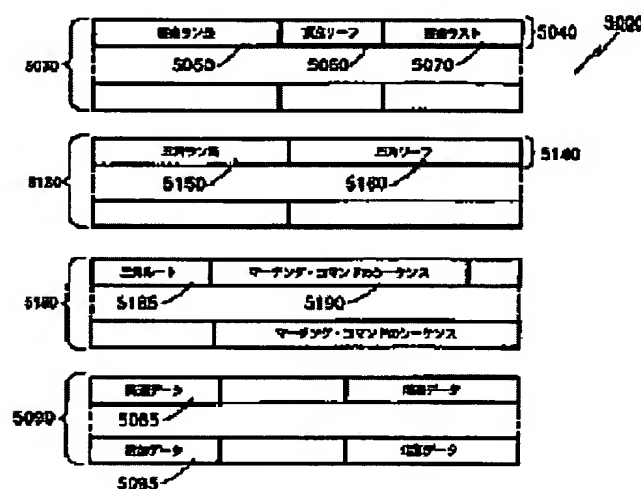
METHOD AND SYSTEM FOR COMPRESSING AND DECOMPRESSING GEOMETRIC MODEL, AND DATA STRUCTURE

Patent number: JP10079049
Publication date: 1998-03-24
Inventor: JAREKU JAROSUROO ROMAN ROSHINATSUKU; GABRIEL TAUBIN
Applicant: INTERNATL BUSINESS MACH CORP <IBM>
Classification:
 - international: G06T17/00; G06F17/50; G06T15/00
 - european:
Application number: JP19970190146 19970715
Priority number(s):

Abstract of JP10079049

PROBLEM TO BE SOLVED: To provide a method and system for compressing, storing, transferring and decompressing a simple triangular mesh.

SOLUTION: A computer system stores and transfers the simple triangular mesh. Data structure 5000 to be used for the computer expresses the simple triangular mesh in an n-th dimensional space. The data structure 5000 is provided with the table of a vertex run, the table of a triangular run and a marching record 5180 giving connected information of a triangular mesh. In addition, data structure 5000 is provided with a related data record including geometric information of the triangular mesh. The table of the triangular run and the marching record 5180 include information describing a method constituting the triangular mesh (namely the vertexes of a polygon and a boundary edge). The table of the vertex run describes a vertex spanning tree giving connected information added for constituting the triangular mesh from the polygon. Related data record decides the accurate position of a spatial triangular mesh. The system also includes method preparing this data structure from the simple triangular mesh, transferring this information between computers, compressing transferred information and decompressing it.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-79049

(43) 公開日 平成10年(1998) 3月24日

(51) Int.Cl. ⁸	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 T 17/00			G 0 6 F 15/62	3 5 0 A
G 0 6 F 17/50			15/60	6 0 1 C
G 0 6 T 15/00				6 0 6 B
				6 2 2 A
			15/72	4 5 0 A
審査請求 未請求 請求項の数38 O L (全 31 頁)				

(21) 出願番号 特願平9-190146

(22) 出願日 平成9年(1997) 7月15日

(31) 優先権主張番号 08/688572

(32) 優先日 1996年7月30日

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州アーモンク (番地なし)

(74) 代理人 弁理士 坂口 博 (外1名)

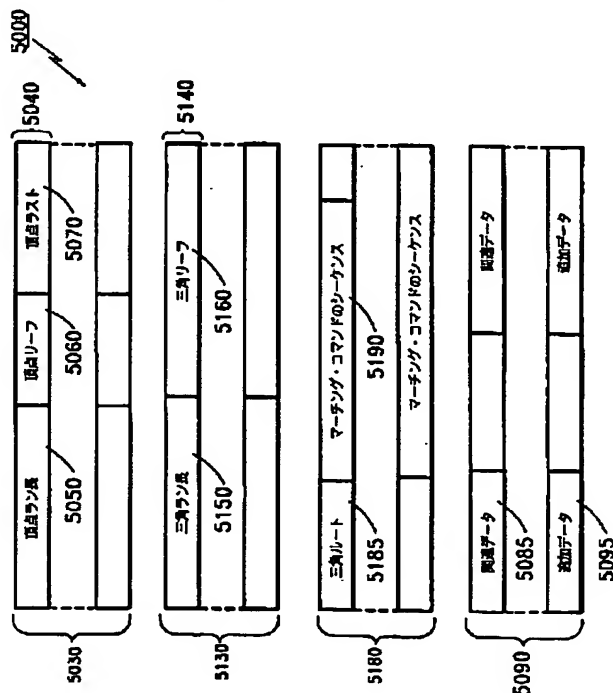
最終頁に続く

(54) 【発明の名称】 幾何モデルを圧縮し圧縮解除する方法、システム及びデータ構造

(57) 【要約】

【課題】 単純三角メッシュの圧縮、保存、転送、圧縮解除を行う方法及びシステムを提供する。

【解決手段】 コンピュータ・システムにより、圧縮された単純三角メッシュを保存し転送する。コンピュータに用いられるデータ構造は単純三角メッシュをn次元空間で表す。データ構造には、頂点ランのテーブル、三角ランのテーブル、三角メッシュの連結性情報を与えるマーチング・レコードを持つ。またデータ構造は、三角メッシュの幾何情報を含む関連データ・レコードを持つ。三角ランのテーブルとマーチング・レコードには、三角メッシュを(従って多角形の頂点と境界エッジを)構成する方法を記述した情報がある。頂点ランのテーブルは、多角形から三角メッシュを構成するため追加される連結性情報を与える頂点スパニング・ツリーを記述する。関連データ・レコードは、空間の三角メッシュの正確な位置を決定する。システムはまた、このデータ構造を単純三角メッシュから作成し、この情報をコンピュータ間で転送し、転送された情報を圧縮し圧縮解除する方法を含む。



【特許請求の範囲】

【請求項1】 n 次元空間の点の集合を表すコンピュータ・メモリのデータ構造であって、ノードを持ち、各ノードが正規ノード、分岐ノード、またはリーフ・ノードであるルート付きツリーを記述し、前記ルート付きツリーの各ノードは、 n 次元空間の点を表し、またレコードを持ち、各レコードは前記ルート付きツリーのラン、すなわちリーフ・ノードか分岐ノードである第1ノードと、リーフ・ノードか分岐ノードである最終ノードを持つランを表し、該ランは、前記第1ノードと前記最終ノードの間の正規ノードを連結し、前記ランの連続したノードのペアそれぞれの連結部はエッジであり、各レコードは、前記ランのエッジ数で表されるラン長フィールド、前記ランが前記リーフ・ノードで終わるかどうを示すリーフ・フィールド、及び前記レコードによって表されるランが、前記ルート付きツリーと同じ前記第1ノードを持つ最後のものかどうかを示すラスト・ラン・フィールドを持ち、前記レコードは、ルート・ノードに対してツリー走査順序で与えられる、前記ランのテーブル、を含む、データ構造。

【請求項2】関連データ・フィールドを含み、各関連データ・フィールドは補正項であり、各補正項は、現在の点の実際の位置と現在の点の予測位置の差であり、現在の点は、カレント・ノードと呼ばれるノードの1つと関連付けられた、請求項1記載のデータ構造。

【請求項3】前記現在の点の予測位置は、現在の点の親元点を使用する予測法により決定され、該親元点は、前記カレント・ノードから前記ルート・ノードへの固有パスにあるノードに対応する、請求項2記載のデータ構造。

【請求項4】前記関連データ・フィールドは固定小数点形式である、請求項1記載のデータ構造。

【請求項5】前記関連データ・レコードは圧縮される、請求項1記載のデータ構造。

【請求項6】前記関連データ・レコードは、ハフマン・エンコーディング、算術コーディング、シャノン・ファノ・エリヤス・コーディング、レンペル・ジブ・コーディングのいずれか1つにより圧縮される、請求項5記載のデータ構造。

【請求項7】点位置をコンピュータの記憶装置に保存するために用いられる、請求項1記載のデータ構造。

【請求項8】コンピュータから通信リンクを通して送られる、請求項1記載のデータ構造。

【請求項9】通信リンクを通してコンピュータにより受信される、請求項1記載のデータ構造。

【請求項10】前記点は、 n 次元幾何形状の頂点を表し、各ツリー・エッジは、前記幾何形状の形状エッジに対応する、請求項1記載のデータ構造。

【請求項11】 n 次元空間で単純三角メッシュを表すデ

ータ構造であって、

頂点ノードを持ち、各頂点ノードが正規頂点ノード、分岐頂点ノード、またはリーフ頂点ノードであり、前記単純三角メッシュの頂点に関連付けられた頂点スパニング・ツリーを記述する頂点ランのテーブルであって、頂点ラン・レコードを持ち、各頂点ラン・レコードは前記頂点スパニング・ツリーの頂点ランを表し、リーフ頂点ノードか分岐頂点ノードである第1頂点ノードと、リーフ頂点ノードか分岐頂点ノードであるラスト頂点ノードとを持ち、前記第1頂点ノードとラスト頂点ノードとの間の正規頂点ノードを連結し、そこで、連続した頂点ノードのペアそれぞれの間の連結部は頂点ツリー・エッジであり、各頂点ラン・レコードは頂点ランの頂点ツリー・エッジ数により与えられる頂点ラン・フィールドの長さ、頂点ラン・レコードにより表される頂点ランのラスト頂点ノードがリーフ頂点ノードかどうかを示す頂点リーフ・フィールド、及び頂点ランが、同じ前記第1頂点ノードを共有するラスト頂点ランかどうかを示す頂点ラスト・ラン・フィールドを持ち、前記頂点ランは、頂点ルート・ノードに対する前記頂点スパニング・ツリーの走査順序により与えられる、前記頂点ランのテーブルと、

三角ノードを持ち、各三角ノードは、正規三角ノード、分岐三角ノード、またはリーフ三角ノードであり、前記単純三角メッシュの三角形に関連付けられた三角スパニング・ツリーを記述する、三角ランのテーブルであって、三角ラン・レコードを持ち、各三角ラン・レコードは前記三角スパニング・ツリーの三角ランを表し、前記三角ランは、リーフ三角ノードか分岐三角ノードである第1三角ノードと、リーフ三角ノードか分岐三角ノードであるラスト三角ノードとを持ち、前記第1三角ノードと前記ラスト三角ノードとの間の正規三角ノードを連結し、そこで、連続した三角ノードのペアそれぞれの間の連結部は三角ツリー・エッジであり、各三角ラン・レコードは、前記三角ラン・レコードで表される三角ランの三角ツリー・エッジ数により与えられる三角ラン長フィールドと、前記三角レコードで表される三角ランの前記ラスト三角ノードがリーフ三角ノードかどうかを示す三角リーフ・フィールドとを持ち、前記三角ランは、三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序により与えられる三角ランのテーブルと、三角ルート・フィールドと、マーチング・コマンドのシーケンスとを持ち、前記三角ルート・フィールドは、前記三角スパニング・ツリーの三角ルート・ノードに関連付けられた三角形を構成する方法を記述し、マーチング・コマンドの各シーケンスは、前記三角ランの1つから三角形を構成する方法を示すため、前記三角ランの左ラン境界または右ラン境界に沿った次の頂点に進み、前記マーチング・コマンドは、前記三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序により与え

られるマーチング・レコードと、
関連データ・フィールドを持ち、各関連データ・フィールドは、前記単純三角メッシュの頂点の1つの位置に関する情報を持ち、前記関連データ・フィールドは、前記頂点ルート・ノードに対する前記頂点スパニング・ツリーの走査順序により与えられる、関連データ・レコードと、
を含む、データ構造。

【請求項12】前記頂点スパニング・ツリーの走査順序は、深さ優先、幅優先、前順、後順のいずれか1つを含む、請求項11記載のデータ構造。

【請求項13】各関連データ・フィールドは補正項であり、各補正項は、現在の頂点の実際の位置と現在の頂点の予測位置の差であり、現在の頂点は、カレント・ノードと呼ばれるノードの1つと関連付けられた、請求項11記載のデータ構造。

【請求項14】前記現在の頂点の予測位置は、現在の頂点の親元の頂点を使用する予測法により決定され、親元頂点は、現在の頂点ノードから頂点ルート・ノードへの固有パスにあるノードに対応する、請求項13記載のデータ構造。

【請求項15】前記関連データ・フィールドは固定小数点形式である。請求項11記載のデータ構造。

【請求項16】前記関連データ・レコードは圧縮される、請求項11記載のデータ構造。

【請求項17】前記関連データ・レコードは、ハフマン・エンコーディング、算術コーディング、シャノン・ファノー・エリヤス・コーディング、レンペル・ジブ・コーディングのいずれか1つにより圧縮される、請求項16記載のデータ構造。

【請求項18】前記三角スパニング・ツリーの走査順序は、前記三角ラン・テーブルの三角ラン・レコードの順序により決定される、請求項11記載のデータ構造。

【請求項19】前記三角スパニング・ツリーの走査順序は、深さ優先、幅優先、前順、後順のいずれか1つを含む、請求項11記載のデータ構造。

【請求項20】前記マーチング・レコードは圧縮される、請求項11記載のデータ構造。

【請求項21】前記マーチング・レコードは、ハフマン・エンコーディング、算術コーディング、シャノン・ファノー・エリヤス・コーディング、レンペル・ジブ・コーディングのいずれか1つにより圧縮される、請求項20記載のデータ構造。

【請求項22】前記マーチング・コマンドはそれぞれビットである、請求項11記載のデータ構造。

【請求項23】追加データ・レコードを含み、追加される各関連データ・レコードは、追加データ・フィールドを持ち、前記追加データ・フィールドはそれぞれ追加情報を持つ、請求項11記載のデータ構造。

【請求項24】前記追加情報は、法線、色及びテキスト

ャ座標を含む、請求項23記載のデータ構造。

【請求項25】前記単純三角メッシュをコンピュータの記憶装置に保存するために用いられる、請求項11記載のデータ構造。

【請求項26】コンピュータから通信リンクを通して送られる、請求項11記載のデータ構造。

【請求項27】通信リンクを通してコンピュータにより受信される、請求項11記載のデータ構造。

【請求項28】複数の連結成分を持つ n 次元空間で前記単純三角メッシュの連結成分を表す、請求項11記載のデータ構造。

【請求項29】幾何モデルを圧縮し圧縮解除する1つ以上のコンピュータ・システムであって、各コンピュータ・システムは、
プロセスを実行する中央処理装置と、
メモリと、

前記メモリに存在し、それぞれが正規頂点ノード、分岐頂点ノード、またはリーフ頂点ノードであり、メッシュの頂点に関連付けられた頂点ノードを持つ頂点スパニング・ツリーを記述する頂点ランのテーブルであって、頂点ラン・レコードを持ち、各頂点ラン・レコードは、前記頂点スパニング・ツリーの頂点ランを表し、前記頂点ランはリーフ頂点ノードか分岐頂点ノードである第1頂点ノードと、リーフ頂点ノードか分岐頂点ノードであるラスト頂点ノードとを持ち、前記第1頂点ノードと前記ラスト頂点ノードとの間の正規頂点ノードを連結し、そこで、連続した頂点ノードのペアそれぞれの間の連結部は、頂点ツリー・エッジであり、各頂点ラン・レコードは、前記頂点ランの頂点ツリー・エッジ数で与えられる頂点ラン長フィールド、前記頂点ラン・レコードで表される頂点ランの前記ラスト頂点ノードがリーフ頂点ノードかどうかを示す頂点リーフ・フィールド、及び前記頂点ランが、同じ前記第1頂点ノードを共有する前記ラスト頂点ランであるかどうかを示す頂点ラスト・ラン・フィールドを持ち、前記頂点ランは、頂点ルート・ノードに対する前記頂点スパニング・ツリーの走査順序により与えられる、頂点ラン・テーブルと、
前記メモリに存在し、それぞれが正規三角ノード、分岐三角ノード、またはリーフ三角ノードであり、前記単純三角メッシュの三角形に関連付けられた三角ノードを持つ三角スパニング・ツリーを記述する、三角ランのテーブルであって、三角ラン・レコードを持ち、各三角ラン・レコードは前記三角スパニング・ツリーの三角ランを表し、前記三角ランはリーフ三角ノードか分岐三角ノードである第1三角ノードと、リーフ三角ノードか分岐三角ノードであるラスト三角ノードとを持ち、前記ランは、前記第1三角ノードと前記ラスト三角ノードの間の正規三角ノードを連結し、そこで、連続した三角ノードのペアそれぞれの間の連結部は、三角ツリー・エッジであり、各三角ラン・レコードは、前記三角ラン・レコー

ドで表される三角ランの三角ツリー・エッジ数で与えられる三角ラン長フィールドと、前記三角レコードで表される三角ランのラスト三角ノードが、リーフ三角ノードかどうかを示す三角リーフ・フィールドを持ち、前記三角ランは、三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序により与えられる、三角ラン・テーブルと、

前記メモリに存在し、三角ルート・フィールドと、マーチング・コマンドのシーケンスを持ち、前記三角ルート・フィールドは、前記三角スパニング・ツリーの三角ルート・ノードに関連付けられた三角形を構成する方法を記述し、マーチング・コマンドの各シーケンスは、前記三角ランの1つから三角形を構成する方法を示すために、前記三角ランの左ラン境界または右ラン境界に沿った次の頂点に進み、前記マーチング・コマンドは、前記三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序で与えられる、マーチング・レコードと、を含む、システム。

【請求項30】2つ以上のコンピュータ・システムが通信リンクで接続された、請求項29記載のシステム。

【請求項31】前記通信リンクは、インターネット、イントラネット、広域ネットワーク、ローカル・エリア・ネットワーク、無線周波数リンク、赤外線リンク及びシリアル通信リンクのいずれかを含む、請求項30記載のシステム。

【請求項32】前記頂点ラン・テーブル、三角ラン・テーブル及びマーチング・レコードは、第1コンピュータ・システムから、前記通信リンクで接続された第2コンピュータ・システムへ転送される、請求項30記載のシステム。

【請求項33】前記頂点ラン・テーブル、三角ラン・テーブル及びマーチング・レコードを、前記幾何モデルから作成する圧縮プロセスを含む、請求項29記載のシステム。

【請求項34】前記頂点ラン・テーブル、三角ラン・テーブル及びマーチング・レコードから前記幾何モデルを作成する圧縮解除プロセスを含む、請求項29記載の装置。

【請求項35】頂点スパニング・ツリーと三角スパニング・ツリーを幾何モデルから作成するステップと、前記頂点スパニング・ツリーを頂点ランのテーブルで記述するステップと、

前記三角スパニング・ツリーを三角ランのテーブルで記述するステップと、

前記頂点ラン・テーブルに従って前記三角スパニング・ツリーを走査しながら、前記三角スパニング・ツリーで三角形を構成する方法を定義するマーチング・レコードを設定するステップと、

を含む、圧縮方法。

【請求項36】前記頂点スパニング・ツリーと前記三角

スパニング・ツリーは、

前記幾何モデルのエッジにコストを割当てるステップと、

前記頂点スパニング・ツリーを総コストが最小のスパニング・ツリーと定義するステップと、

前記三角スパニング・ツリーを前記頂点スパニング・ツリーに含まれるエッジで作成するステップと、

で作成される、請求項35記載の圧縮方法。

【請求項37】前記頂点スパニング・ツリーと前記三角スパニング・ツリーは、

a. 前記幾何モデルの現在の頂点の集合を選択するステップと、

b. 現在の頂点を含み、三角化された現在の層として走査されていない全ての三角形を選択するステップと、

c. 選択されないエッジは分離エッジであり、層エッジとして割当てられておらず、前記現在の頂点を少なくとも1つ含む、全てのエッジを選択するステップと、

d. 前記層エッジが、前記三角スパニング・ツリーにループを作らない場合には、前記層エッジそれぞれを前記三角スパニング・ツリーに割当て、そうでない場合は前記頂点スパニング・ツリーに割当てるステップと、

e. 前記層エッジが前記三角スパニング・ツリーにループを作らない場合は、前記分離エッジそれぞれを前記三角スパニング・ツリーに割当て、そうでない場合は前記頂点スパニング・ツリーに割当てるステップと、

f. 前記分離エッジを現在の頂点の集合として選択するステップと、

g. エッジがなくなるまでステップa乃至fを繰り返すステップと、

h. 割当てられた対応する前記エッジを使用して前記頂点スパニング・ツリーと前記三角スパニング・ツリーを作成するステップと、

を含む、請求項35記載の圧縮方法。

【請求項38】頂点位置を再構成するステップと、

境界ループを構成するステップと、

Y境界ループ・ノードの相対索引を決定するステップと、

三角ストリップを再構成しリンクするステップと、

を含む、圧縮解除方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はコンピュータ・グラフィックスに関し、特に3次元幾何モデルの圧縮転送と圧縮保存に関する。

【0002】1996年1月16日出願の係属中の米国特許出願第010045号"Compression of Geometric Models Using Spanning Trees"参照。

【0003】

【従来の技術】メカニカルCAD（コンピュータ支援設計）やアニメーションの分野でのモデリング・システム

は、幾何学の対象分野を自由局面にまで広げているが、多面体モデルは、製造、建築、地理情報システム、地学、エンターテインメント等の産業で用いられる基本的な3D表現にとどまっている。多面体モデルは特にハードウェア支援レンダリングに効果的である。ハードウェア支援レンダリングは、複雑なCADモデルを伴うビデオ・ゲーム、バーチャル・リアリティ（仮想現実）、フライスルー、電子モックアップ（mock-up）等のアプリケーションには重要である。

【0004】3D形状の圧縮については、画像やビデオの圧縮に比べて研究分野でも、データ交換規格に携わる委員会等でも、あまり注意が払われていない。しかし、この圧縮の必要性は、次の3つの理由から急激に拡大している。1) 業務用CADモデルがますます複雑になっているため、こうしたモデルに必要なメモリや補助記憶装置のコストが大幅に増加している。2) 共同で行う設計、ゲーム、短時間でのプロトタイプ作成、バーチャル・インタラクション等を目的にしたネットワークでの3Dモデルの配付は、使用できる帯域幅によって大きな制約を受ける。3) ハイレベルのハードウェア・アダプタのグラフィクス性能は、モデル全体を保存するには不十分なオンボード・メモリや、データ転送のボトルネックにより制限される。

【0005】多面体の表面を簡単に、効率的に三角化する方法は知られているので、三角メッシュで定義された幾何モデルを考慮すれば充分である。三角メッシュは、 n 次元ベクトルであるその頂点（幾何）の位置、各三角とそれを支える頂点との関連（連結性）、及び3D幾何には影響を与えないが、陰影付けの方法には影響を与える色、法線及びテキストチャ座標（測光）により定義される。

【0006】従来の技術は、3D幾何の圧縮に関して、3つのカテゴリに分けることができる。簡素化、幾何エンコーディング、及び連結性エンコーディングである。

【0007】多面体の簡素化法では、メッシュの頂点数を減らすために、モデルの連結性の変更され、簡素化により生じる誤差を最小限にするために、残りの頂点の位置を調整することもできる。こうした手法は、オーバーサンプリングのメッシュについてグラフィクスやデータの縮小を加速するために、複数のLOD（Levels Of Detail、詳細レベル）を生成することが目的である。これらの手法は、損失の多い圧縮について考慮されるが、モデルの正確な連結性にアクセスする必要のあるアプリケーションには適さない。実際、簡素化法は、ここで述べる圧縮法とは対照的（orthogonal）である。幾何圧縮は各詳細レベルに適用できるからである。

【0008】幾何エンコーディング法は、損失の多い圧縮が損失のない圧縮により、頂点位置、或いは色、法線及びテキストチャ座標に関連した幾何データに必要な記憶域を減らす。汎用のバイナリ圧縮アルゴリズムを幾何デ

ータ・ストリップに適用することは、近最適解に近づくことである。Michael Deeringによる"Geometric Compression", Computer Graphics (Proc. SIGGRAPH), pages 13-20, August 1995には、形状を単位立方体に正規化し、頂点座標を固定長整数に丸める幾何圧縮法が説明されている。丸め操作は、失われた情報の量を制御するものである。

【0009】連結性エンコーディング法は、多面体や三角メッシュの一般的な多くの表現に特有の冗長性を減らすとするものである。例えば、連結性エンコーディング法は、頂点が V 個で三角形が T 個の三角メッシュの三角形を表すのに必要なビット数を減らすとする。

【0010】端的な例をあげると、頂点が常に正規の2Dグリッドに構成されている場合、三角メッシュはグリッドの行列によって完全に定義できる。正規グリッドは、地理情報システムの地勢モデリングや、一様にテッセレート（tesselated）され、トリミングされていない正方パラメトリック・パッチのレンダリングに適している。しかし、CAD、エンターテインメント、その他のアプリケーションに見られるより一般的な3D形状のモデリングには適していない。

【0011】もう1つ端的な例をあげると、頂点が V 個で三角形が T 個の三角メッシュの頂点位置は、アレイ、頂点位置アレイで表せ、各三角形は頂点位置アレイに対する3つの索引によって表せる。この解決法では、形状的制限がメッシュに課されることはないが、三角形当たり3つのアドレスを保存する必要がある（代表的な三角メッシュで頂点当たり約6アドレス）。モデルが1000頂点に限られる場合でも、この手法では、連結性情報だけで頂点当たり60ビットが使われる。

【0012】グラフィクスAPI（アプリケーション・プログラミング・インターフェイス）に用いられる三角ストリップ（J. Neider, T. Davis及びM. WooによるOpenGL Programming Guide, Addison-Wesley, 1993で説明されている）は、前の2つの頂点と新しい頂点が組み合わせられ、現在のストリップの新しい三角形が暗示的に定義される遠い妥協案である。三角ストリップが意味を持つのは、長いストリップを作成できる場合だからである。長いストリップは、計算幾何に突きつけられた課題である。更に頂点は、同じ三角ストリップの一部としてか、異なる2つのものの一部として、平均で2回用いられるので、OpenGLで三角ストリップを使用するには、ほとんどの頂点を何回も送信する必要がある。こうした冗長性は、スワップ操作がないために更に多くなる。

【0013】圧縮解除時に、全ての頂点の位置をランダム・アクセスに使用できる圧縮法として、三角ストリップを応用するには、未だ三角形当たり1つの頂点参照、ストリップ当たり2つの頂点参照、ストリップの数と長さに関する記録情報、及び次の三角形の基準とする前の三角形の開き側を示す三角毎の追加情報ビット（このビ

ットはGLのSWAP操作に等しい)を保存する必要がある。

【0014】Deeringの方法では、モデルの全ての頂点に対するランダム・アクセスを行う代わりに、先に使われた頂点ノード16個を保存するのにスタック・バッファが用いられる。この解決法は、オンボード・メモリがかなり制限されたアダプタには適している。Deeringはまた、三角ストリップ構文を一般化するために、次の頂点の使用方法についてより一般的な制御を行い、現在の頂点をスタックに一時的に追加し、スタック・バッファの16頂点ノード1つを再使用できるようにする。この連結性情報の保存コストは、頂点をスタック・バッファにプッシュするかどうかを示す頂点当たり1ビット、カレント・ストリップを続ける方法を示す三角形当たり2ビット、新しい頂点を読むか、またはスタック・バッファから頂点を使用するかを示す三角形当たり1ビット、及びスタック・バッファから頂点を選択するための4ビットのアドレスであり、これは古い頂点が再使用される度に必要のコストである。頂点当たり2つの三角形を想定すると、総コストは頂点当たり約11ビットである。本出願者が知る限り、Deeringの一般化された三角メッシュ構文を使用して、一般メッシュの良好な走査経路を系統的に作成するアルゴリズムは、まだ開発されていない。メッシュの単なる走査では、分離した三角形または小さいラン(run)が多くなることが考えられ、これはつまり頂点のかなりの部分が2回以上送られ、従って三角形当たりのビット数が増えることを意味する。

【0015】連結性エンコーディングを考慮した従来の技術の方法は、どれも三角メッシュの元の連結性を保存しない。これらの方法は全て三角メッシュを、三角形の連結されたサブセットに分割する。2つ以上の部分によって共有される頂点が何回も表現される。これは、圧縮法がCPUとコンピュータのグラフィクス・アダプタとの間の通信帯域幅をうまく利用するために用いられるものであるなら、問題にはならないかもしれないが、こうした連結性の変化は、元のメッシュの形状的性質を保存する必要のある他のアプリケーションでは重大な欠点になる。

【0016】図1と図2は、従来の技術のツリーの例である。従来の技術("R. E. Tarjan, "によるData Structures and Network Algorithms, SIAM, 1983に説明がある)は、ツリー1000がノードと、それぞれがノードのペアをつなぐエッジで構成されることを認識している。1つのエッジ1015で別のノード1030に連結されたノード1010は、リーフ・ノードである。2つのエッジ(1025、1035)で他の2つのノード(1030、1050)に連結されたノード1020は、正規ノードである。3つ以上のエッジ(1015、1035、1045)で対応するノードに連結されたノード1030は分岐ノードである。ツリーはランのセッ

トとして記述することができる。ランとはノードのシーケンスであり、連続したノードのペアがそれぞれエッジで連結されている。このシーケンスには、リーフ・ノードか分岐ノードである第1ノード、正規ノードである0以上の中間ノード、及びリーフ・ノードか分岐ノードである最終ノードがある。例えばリーフ・ノード1040、正規ノード1050、1020及び分岐ノード1030により、ラン(1040、1050、1020、1030)が定義される。

【0017】1100はルート付きツリーで、これは、ルート・ノードと識別される1つのノード1110を持つツリーである。ルート・ノードはエッジで連結されたノードの各ペア間の親子関係を決定する。ルート・ノードはまた、各ランの最初と最後のノードを決定する。ルート・ノードは部分的には、ルート付きツリーの走査の順序を決定する。走査の順序を完全に指定するには、ランを更に共通第1ノード・ランのセットに分割する必要がある。各セットは、共通第1ノードを共有する全てのランで構成される。また共通第1ノード・ランの各セット内のランに順序を割当てて必要がある。例えばルート付きツリー1100のルートとしてノード1110を選択すると、ノード1120と1150はそれぞれ、ラン(1120、1130、1140、1150)の最初と最後のノードになり、ノード1120、1180はそれぞれ、ラン(1120、1160、1170、1180)の最初と最後のノードになる。しかし、これら2つのランのどちらが最初に走査されるかは、ツリーのルートとしてノード1110を選択したことからは決定されず、明示的に指定する必要がある。

【0018】図3、図4、図5、図6及び図7は、三角メッシュの従来の技術の例である。三角メッシュ2100は、頂点2110、エッジ2120、及び三角形2130で構成される。

【0019】三角メッシュのパスは、連続した頂点の各ペアがエッジによって連結されるように、繰り返されない頂点のシーケンスである。パスの最初と最後の頂点は、パスによって連結されると言われる。例えば頂点2140と2165は、頂点2140、2150、2160及び2165によって形成されるシーケンスで定義されたパスにより連結される。三角メッシュはその頂点の全てのペアがパスで連結される場合に連結される。例えば三角メッシュ2100は連結されており、三角メッシュ2200は連結されていない。三角メッシュが連結されていないとき、それは2つ以上の連結成分で構成され、連結成分はそれぞれ元の三角メッシュの頂点、エッジ及び三角形のサブセットで構成され、パスで連結されている場合は、同じ連結成分に属する三角メッシュの2つの頂点に連結された三角メッシュである。例えば三角メッシュ2200は、2つの連結成分、三角メッシュ2210及び2220で構成される。頂点2230と22

40は、パスで連結できるので同じ連結成分に属する。頂点2230、2250はそれらをつなぐパスが存在しないので属する連結成分が異なる。

【0020】三角メッシュのエッジは、正確に1つの三角形に属する場合は境界エッジ、正確に2つの三角形で共有される場合は内部エッジ、そして3つ以上の三角形で共有される場合は単一エッジである。例えば三角メッシュ2300のエッジ2350は、境界エッジ、三角メッシュ2100のエッジは全て内部エッジ、三角メッシュ2300のエッジ2340は単一エッジである。これは三角形2310、2320、2330で共有されているからである。(境界エッジと内部エッジは正規エッジとも呼ばれることに注意されたい。)三角メッシュの頂点は、与えられた頂点を除いて頂点を持つ、三角メッシュの全ての三角形の頂点のセットを再び順序付けして、1つのパスを定義できる場合は正規頂点である。頂点が正規頂点でなければ、それは単一頂点である。例えば三角メッシュ2100の頂点は全て正規であり、三角メッシュ2400の頂点2410は単一である。これは頂点2410を除いて、頂点2410に付随する全ての三角形の頂点を再順序付けして1つのパスを形成する術がないからである。三角メッシュは、境界エッジがあるなら境界を持つ。例えば三角メッシュ2100と2200には境界がなく、三角メッシュ2300と2400には境界がある。三角メッシュは、単一頂点も単一エッジもない場合はマニフォールド(manifold)である。三角メッシュがマニフォールドでなければそれは非マニフォールドである。例えば三角メッシュ2100と2200はマニフォールドであるが、三角メッシュ2300は非マニフォールドである。

【0021】三角メッシュの三角形の向きは、巡回置換を法とした三角形のツリー頂点の順序である。従って、三角メッシュの三角形には2つの向きが考えられる。例えば、頂点2110、2170、2180で構成される三角形2130は順序(2110、2170、2180)と(2180、2170、2110)である。しかし順序(2110、2170、2180)と(2180、2110、2170)は、ある巡回置換によって関係付けられるので同じ三角形の向きを定義する。

【0022】三角メッシュのエッジの向きは、エッジの2つの頂点について考えられる2つの順序の1つである。例えば三角メッシュ2100のエッジ2120の2つの向きは、頂点2110と2170の順序(2110、2170)と(2170、2110)によって定義される。エッジの2つの向きは対向していると言われる。三角形の向きは、付随する3つのエッジに一貫した向きを誘発する。例えば向き(2110、2170、2180)は、三角形2130の3つのエッジに一貫した向き(2110、2170)、(2170、2180)及び(2180、2110)を誘発する。三角メッシュ

の内部エッジそれぞれについて、エッジに付随する2つの三角形が、共通のエッジ上に対向する向きを誘発するような形で、三角メッシュの各三角形について向きを選択できる場合、マニフォールド三角メッシュは配向可能である。マニフォールド三角メッシュが配向可能でなければ、それは配向不能である。非マニフォールド三角メッシュは全て配向不能である。例えば三角メッシュ2100は配向可能で、三角メッシュ2500は配向不能である。

【0023】頂点V個、エッジE個、三角形T個の三角メッシュのオイラー特性は、数、 $\chi = V - E + F$ である。従来の技術("W. S. Massey"によるAlgebraic Topology: An Introduction, Harcourt, Brace & World, 1967に説明)では、同じオイラー特性の2つの連結マニフォールド三角メッシュは両方が配向可能であるか、両方が配向不能である場合には、相互に連続して変形可能であることが知られている。

【0024】従来の技術("Foley"らによるComputer Graphics: Principles and Practice, Addison-Wesley, 1990に説明)では、普通、頂点V、三角Tの三角メッシュは、頂点位置アレイと三角アレイで表される。三角メッシュの各頂点の位置は、n個の浮動小数点座標により頂点位置アレイで表される。三角メッシュの各三角形は、頂点位置アレイに対する3つの索引によって三角アレイで表される。

【0025】この開示の目的からは、単純三角メッシュは、連結、配向済み、マニフォールド、無境界、そしてオイラー特性2のn次元頂点を持つ三角メッシュである。例えば三角メッシュ2100は3次元空間の単純三角メッシュである。例えば、どのような単純三角メッシュも、球面を三角化して連続的に変形することによって得られる。

【0026】

【発明が解決しようとする課題】本発明の目的は、単純三角メッシュの圧縮、保存、転送、圧縮解除を目的にした、改良されたシステム及び方法である。

【0027】本発明の目的は、単純三角メッシュの連結性情報、すなわち単純三角メッシュのどの頂点が、圧縮された形で情報を失うことなくメッシュの三角面を形成するかに関する情報を表す、改良されたシステム及び方法である。

【0028】本発明の目的は、単純三角メッシュの頂点位置、法線、色及びテキスチャ座標を圧縮された形で表す、改良されたシステム及び方法である。

【0029】

【課題を解決するための手段】本発明は、単純三角メッシュの圧縮、保存、転送及び圧縮解除を目的にしたコンピュータ・システム及び方法である。

【0030】コンピュータは単純三角メッシュを圧縮された形で表すデータ構造を用いる。データ構造は単純三

角メッシュの連結性情報を記述した三角形ランのテーブル、1つのマーチング・レコード及び頂点ランのテーブルを持つ。またデータ構造には、単純三角メッシュの頂点の位置を記述した情報を持つデータ・レコードが任意に関連付けられる。データ構造はまた、データ・レコードを追加することができる。追加される各データ・レコードは、単純三角メッシュの色、法線またはテキストチャ座標の1つを記述した情報を持つ。

【0031】単純三角メッシュは、三角化した多角形で表すことができる。多角形の三角形は多角形三角形と呼ばれ、三角化した多角形には内部頂点はなく、多角形の境界は閉じており、三角メッシュの頂点それぞれが多角形の境界にある頂点に対応する。三角形ランのテーブルとマーチング・レコードにより、多角形がどのように三角化されるかが決まる。三角化した多角形から三角メッシュを再生するため、三角化した多角形の境界の頂点（エッジ）が、頂点ランのテーブルで定義される頂点スパニング・ツリーを走査することによって連結される。

【0032】三角メッシュ（表面等）を三角化多角形に変換する（またその逆を行う）ことによって、三角メッシュに関する情報の保存、圧縮、転送及び圧縮解除を効率よく行うことができる。

【0033】三角ランのテーブルは、三角スパニング・ツリーを記述する。三角スパニング・ツリーには三角ノードと三角エッジがある。各三角ノードに多角形三角形が関連付けられる。これら複数の多角形三角形が多角形を三角化する。これにより多角形は多角形頂点と多角形境界を持ち、そこで全ての多角形頂点は多角形境界に位置する。境界エッジとよばれる多角形三角形のエッジのいくつかは、多角形境界に位置し、2つの多角形頂点をつなぐ。内部エッジとよばれる多角形三角形の残りのエッジはそれぞれ、2つの多角形三角形に共有され、多角形の内側にある。

【0034】マーチング・レコードには、三角スパニング・ツリーを走査することによって多角形三角形を（従って多角形頂点と境界エッジを）構成する方法を記述する情報もある。

【0035】頂点ランのテーブルは頂点スパニング・ツリーを記述する。頂点スパニング・ツリーは頂点ツリー・ノードと頂点ツリー・エッジを持つ。頂点スパニング・ツリーは、多角形から単純三角メッシュを構成するために別に連結性情報を提供する。頂点スパニング・ツリーを走査するとき、境界エッジのペアが単純三角メッシュの単一エッジと識別される。従って、頂点スパニング・ツリーを走査することで頂点ツリー・ノードと単純三角メッシュの頂点の間に1対1の対応がとられる。

【0036】単純三角メッシュの連結性情報が確立されると、関連データ・レコードに保持された情報により、単純三角メッシュの頂点位置が決定される。オプションの付加データ・レコードに保持された情報は、三角メッ

シュの色、法線及びテキストチャ座標を決定する。

【0037】本発明はまた、当該データ構造で表された単純三角メッシュを圧縮解除し、当該データ構造で表された、圧縮された単純三角メッシュをコンピュータ間で転送するために、単純三角メッシュを当該データ構造で表すことによって圧縮するシステム及び方法を提供する。

【0038】

【発明の実施の形態】コンピュータ・グラフィクスでは、幾何オブジェクトや幾何モデルは三角メッシュによって表すことができる。三角メッシュは更に単純三角メッシュで表せる。ここでは、これらのメッシュの例を図11乃至図16及び図19乃至図22に示してある。図18の圧縮プロセス8000で、これらのメッシュは図12に示してある方法でエッジに沿って切断して、三角メッシュ（幾何オブジェクト、図11等）の三角化した多角形（図15）表現を作成することができる。多角形の三角化は三角ランのテーブルとマーチング・レコードで与えられた情報により定義される。（一般には図10を点の集合を使用した実施例については図8、図9を参照されたい。）三角化された多角形から三角メッシュを再生するのに必要な他の情報は頂点ランのテーブルに保存される。この情報は三角メッシュの頂点に関する位置情報と共に図10に示してある。三角メッシュを再生する圧縮解除プロセスは図28に示してある。

【0039】 n 次元空間で点の集合（三角メッシュのサブセット、すなわちその頂点）を表す他の実施例の1つでは、図8、図9に示したデータ構造で記述される情報が用いられる。空間の点の集合を扱うこの例で点はエッジで連結され、これらの点とエッジはツリーで表すことができる。ツリーの各ノードは点の1つに関連付けられる。ツリーを正規の順序で走査することで点はその元の位置の空間に置くことができる。図8には、ツリーを定義するデータがあり、図9には、点が空間で相互にどのような位置関係にあるかを定義するデータがある。図28のプロセスは、これらのデータ構造に対して実行され、ツリーから空間の点を再生する。

【0040】図8は、ルート付きツリーを表すデータ構造3000のブロック図である。これはランのテーブル3200で構成され、3200はレコード3300で構成される。各レコード3300はルート付きツリーのランを表し、ラン長フィールド3400、リーフ・フィールド3500及びラスト・フィールド3600で構成される。ラン長フィールド3400はランの最初と最後のノードをつなぐエッジの数である。リーフ・フィールド3500はランの最終ノードがリーフ・ノードかどうかを示す。共通第1ノードから始まるランはいくつか考えられる。"ラスト"・フィールド3600はランが共通第1ノードで始まるランの集合の最後のランかどうかを示す。テーブルのレコードの順序はルート・ノードで定義

されるルート付きツリーの走査の順序と、共通第1ノードのランの集合のそれぞれのランに割当てられる順序によって決定される。順序は後述する圧縮プロセスによって定義される。

【0041】図9は、 n 次元空間の点の集合を表すデータ構造4000のブロック図である。ここで集合の各点はルート付きツリーのノードに関連付けられる。データ構造4000はルート付きツリーを表すデータ構造3000と、データ・フィールド4300が関連付けられたもう1つのデータ・レコード4200とで構成され、関連付けられた各データ・フィールドには点の1つの位置に関する情報がある。

【0042】ルート付きツリー内にノードのそれぞれから親子関係で決定されるルート・ノードへの固有パスがある。ルート・ノードの各ノードからの固有パスはノードと、ノードの親元ノードで構成され、連続したノードのペアはそれぞれエッジで連結される。ルート付きツリーのノードが近接していることは、 n 次元空間の対応する点が幾何学的に近接していることを意味することが多いので、好適な実施例では、各点の予測位置を予測するため、ルート付きツリーの各ノードの親元ノードに関連付けられた点の位置も本発明に用いられる。次に、点の予測位置と実際の位置の違いを示す補正項が用いられる(エンコードされる)。ある好適な実施例では関連付けられたデータ・フィールド4300は補正項である。

$$P(\lambda, p_{n-1}, \dots, p_{n-K}) = \lambda_1 p_{n-1} + \dots + \lambda_K p_{n-K}$$

【0046】ここで $\lambda = (\lambda_1, \dots, \lambda_K)$ と K は予測子のパラメータである。 $K=1$ と $\lambda_1=1$ を選択することによって、Deeringが用いたデルタが特別な事例としてカバーされる。より好適な実施例の場合、パラメータ $\lambda_1, \dots, \lambda_K$ は、次のように補正項の累乗の和を最小化するように選択される。

【数3】

$$\sum \|\epsilon_n\|^2 = \sum \|P(\lambda, p_{n-1}, \dots, p_{n-K})\|^2,$$

【0047】これは、三角メッシュの全ての頂点に対応する。従って4300は、ノードをその予測位置からその実際の位置に移すのに使われる補正項、 ϵ_n である。

【0048】好適な実施例で点の位置は量子化される、すなわち固定小数点表示法で最も近い数に切り詰められる。より好適な実施例では、関連データ・フィールド4300はそれぞれ補正項 ϵ_n である。もう1つの好適な実施例では、関連データ・レコードは圧縮される。また別の実施例では、エントロピ・エンコーディング(C. W. Brown及びB. J. ShepherdによるGraphics File Formats, Prentice Hall, 1995に説明)により関連データ・レコードが圧縮される。

【0049】図10は、本発明で開示しているデータ構造5000のブロック図である。これは n 次元空間の単

【0043】例えば図2で、ルート付きツリー1100のノードに n 次元空間の点に関連付けられる場合、ノード1130からルート・ノード1110へのパスは、2つの親元ノード1120と1110で構成される。点1130の位置は、補正項1097を予測位置1090に追加することで得られる。

【0044】各ノードの深さはノードからルートへの固有パスのノードをつなぐエッジの数であり、ルート・ノードの深さは0に等しい。 ρ_n が深さ n のノードに関連付けられた点の位置のとき、 ρ_n は次の和として圧縮解除できる。

【数1】

$$\rho_n = \epsilon_n + P(\lambda, p_{n-1}, \dots, p_{n-K})$$

【0045】ここで ϵ_n は、点に関連付けられた補正項、 P は予測子関数、 $\lambda = (\lambda_1, \dots, \lambda_K)$ と K は親元の数に等しいパラメータ、 $\rho_{n-1}, \dots, \rho_{n-K}$ は、ノードからルート・ノードへの固有パスに沿ったノードの K 個の親元に関連付けられた点の位置である。ルート付きツリーのトップ・ノードには K 個の親元がない場合があるので、ルート・ノードは負の深さの各親元に代わって用いられる。予測子関数は、好適な実施例では次式で定義される線形予測子である。

【数2】

純三角メッシュを表す。データ構造は次のように構成される。

1) 頂点ラン5030のテーブルで記述される頂点スパニング・ツリー。頂点ランのテーブルには頂点ラン・レコード5040があり、各頂点ラン・レコードに、頂点ラン長さフィールド5050、頂点リーフ・フィールド5060及び頂点ラスト・フィールド5070がある。

2) 三角形ランのテーブル5130で記述される三角スパニング・ツリー。三角形ランのテーブルには三角形ラン・レコード5140があり、各三角形ラン・レコードに、三角形ラン長さフィールド5150と三角形リーフ・フィールド5160がある。

3) 三角ルート・フィールド5185と、マーチング・コマンド5190のシーケンスを持つマーチング・レコード5180。三角ルート・フィールド5185は、三角スパニング・ツリーの三角ルート・ノードに関連付けられた三角形を構成する方法を記述し、マーチング・コマンドのシーケンス5190はそれぞれ、A) 三角形ランの左のラン境界、B) 右のラン境界、のいずれかに沿って次の頂点に進むことによって三角形ランの1つから三角形を構成する方法を記述する。

4) 関連データ・フィールド5085を持つ(オプションの) 関連データ・レコード5080。関連データ・レ

コードはそれぞれ、単純三角メッシュの1つの頂点の位置に関する情報を持つ。

5) (オプションの) 追加データ・レコード5090。それぞれ追加データ・フィールド5095を持ち、5095はそれぞれ、単純三角メッシュの色、法線、またはテキスチャ座標の1つのベクトルに関する情報を持つ。

【0050】図11乃至図16は、三角メッシュの成分とデータ構造5000の要素の関係を示す。

【0051】図11は、頂点スパニング・ツリー6100を作成するために選択されたエッジ6105を持つ単純三角メッシュ6000（幾何モデル）である。概略は図13にも示してある。単純三角メッシュは、図12の頂点スパニング・ツリー6100のエッジに沿って切断される。この切断で作成される境界が図14に示してある。図15は三角メッシュを“平坦化”する、すなわち、三角メッシュの全ての頂点と切断エッジを1つの平面に並べることによって作成され、三角化した多角形6500を示す。（これはオレンジの皮をむき取り、皮をテーブルに並べることを想像すれば理解しやすい。）図16は、三角化した多角形6500に三角スパニング・ツリー6690が重ね合わされた6600を示す。データ構造5000は、図18の圧縮プロセス8000により埋められ、より圧縮された形の三角メッシュ6000が表される。

【0052】図28の圧縮解除プロセスは、三角形ランのテーブル5130で記述された三角スパニング・ツリー6690を走査し、マーチング・レコード5180の情報により、三角化した多角形6500が一意に完全に定義される。頂点ラン・テーブル5030で記述された頂点スパニング・ツリー（図13、6300）を走査することで、三角化した多角形6500が“縫い合わせられ”、図11の三角メッシュ6000が再生される。

【0053】本発明で、単純三角メッシュ6000は単純三角メッシュの頂点とエッジで定義されるグラフ上のルート付きツリー、頂点スパニング・ツリー6100により圧縮された形で表される。頂点スパニング・ツリーと、n次元空間の点の集合とみられるメッシュの頂点の集合は、データ構造5000で頂点ラン・テーブル5030と、n次元の点の集合が本発明でランのテーブル3200により表現される方法と同様に関連データ・レコード5080、及び図9のデータ構造4000の関連データ・レコード3100により表される。また追加データ・レコード5090には、色、法線及びテキスチャ座標に関する情報を任意に追加することができる。

【0054】メッシュが頂点ツリー・エッジで切断される場合、頂点スパニング・ツリー6100は、境界ループ・エッジを持つ境界ループになる。図12は、頂点ツリー・エッジでメッシュを切断することによって作成される形状の不連続性を人工的に拡大することによる境界ループ6200の形成を示す。切断の後、各頂点ツリー

・エッジ6105は2つの境界ループ・エッジ6205、6215に対応する。頂点ツリー・エッジ6105と境界ループ・エッジ（6205、6215）の対応は、頂点スパニング・ツリー（6100、6300）により暗示的に記述される。この対応は、後述する圧縮プロセスと圧縮解除プロセスに用いられる。対応は暗示的なので、対応に関する情報を維持する必要はない。本発明では頂点ツリー・エッジはカット・エッジ6105とも呼ばれる。

【0055】図14は、境界ループ6200の表現6400の概略を示す。境界ループ6200には境界ループ・ノード（6401等）がある。（境界ループ（6200、6400）は、三角化した多角形6500の境界でもあることに注意されたい。）頂点スパニング・ツリーの各ノード（6301乃至6304等）は境界ループ・ノードに対応する。例えば頂点スパニング・ツリーの頂点リーフ・ノード6301はそれぞれ、1つの境界ループ・ノード6401に対応する。頂点スパニング・ツリーの頂点正規ノード6302はそれぞれ、2つの境界ループ・ノード6402、6403に対応する。頂点スパニング・ツリーの頂点分岐ノード6304はそれぞれ、3つ以上の境界ループ・ノード6404、6405、6406、6407、6408に対応する。

【0056】圧縮プロセス8000は、頂点スパニング・ツリー6300の頂点走査順序6380を定義するのに対して、圧縮解除プロセス15000は、頂点スパニング・ツリー6300の頂点走査順序6380を用い、三角化した多角形6500を“縫い合わせる”。頂点走査順序6380は、圧縮プロセス8000と圧縮解除プロセス15000が頂点スパニング・ツリー6300の頂点ランを走査する順序を定義する。例えば頂点走査順序6380は、最初に頂点ラン6310を走査し、次に6320、3番目に6330、4番目に6340、5番目に6350を走査する。好適な実施例では、頂点走査順序6380は頂点ラン5040（3300）が圧縮プロセス8000で頂点ラン・テーブル5030（3200）に置かれる順序で表される。

【0057】三角メッシュ6000がカット・エッジで切断されるとき、三角化した多角形6500（内部頂点なし）が得られる。図15は、人工的に平坦化された境界ループ6400で囲まれた三角化した多角形6500を示す。境界ループ・ノード6401、6402、6403、6404、6405、6406、6407及び6408は、三角化多角形6500の多角形頂点6501、6502、6503、6504、6505、6506、6507及び6508に対応する。境界ループ6400は、三角化多角形6500の三角形を図16に示した三角スパニング・ツリー6690として構成する。三角スパニング・ツリー6690は、三角化多角形6500の双対グラフであり、三角ノード（6610、662

0等)と呼ばれる三角スパニング・ツリーの各ノードが、三角化多角形6500の多角形三角形(6611、6621等)に関連付けられる。

【0058】三角スパニング・ツリー6690は、データ構造5000で三角ランのテーブル5130によって表される。三角ラン・テーブルの三角ランのレコードの順序は三角ルート・ノード(6610等)に対する三角形走査順序6680によって決定される。(三角ルート・ノード6610は、ルート付きツリーである三角スパニング・ツリー6690のルート・ノードである。)三角ラン・レコード5140はそれぞれ三角スパニング・ツリーのランを表し、ラン・フィールド長さ5150とリーフ・フィールド5160の2つのフィールドで構成される。これらのフィールドの意味は図8のデータ構造3000と同じであるが、三角スパニング・ツリーはバイナリなので、三角ラン・レコードに”ラスト”・フィールドは不要である。

【0059】三角ラン・テーブル5130で記述される三角スパニング・ツリー6690の構造は、境界ループ6400の境界ループ・ノード(6401乃至6404等)をつなぐことによって三角化多角形6500を三角化する方法を一意に指定することはない。同じ双対グラフ(つまり三角スパニング・ツリー6690)を持つ三角化多角形6500は様々ある。従って、三角化多角形6500を実際に三角化する方法を完全に記述するには、マーチング・レコード5180に保持される情報が必要である。

【0060】三角スパニング・ツリー6690はルート付きツリーである。三角ルート・ノード6610は、マーチング・レコード5180の三角ルート・フィールド5185で表され、マーチング・コマンドはマーチング・レコード5180で表される。

【0061】好適な実施例では、三角ルート・ノード6610は対応する三角形6611で連結された3つ連続した境界ループ・ノード(6514、6504、6524)の第2境界ループ・ノード6504で識別される三角スパニング・ツリー6690の三角リーフ・ノードである。圧縮解除プロセス15000は三角ラン・テーブル5130のレコード5140の順序で定義されるように三角スパニング・ツリー6690を走査することによって、境界ループ6400を多角形三角形(6611等)で埋めて三角化多角形6500を再生する。

【0062】図17は圧縮プロセス8000で、三角ラン・レコード5140に関連付けられたマーチング・コマンドのシーケンス5190それぞれのマーチング・コマンドを作成する方法を示す。図17に示した三角ランの例7000は境界ループ6400の一部が、ランの走査方向7005に対して決定され、左ラン境界7210と右ラン境界7310によりどのように構成されるかを示す。三角ラン7000が走査されると(7005)、

三角形走査順序6680で決定される順序でマーチング・エッジ(7600等)が横切られる。マーチング・エッジは三角化多角形6500の内部エッジである。マーチング・エッジ7600はそれぞれ頂点を三角ランの前のマーチング・エッジ7500と共有する。共有される頂点7700はそれぞれ左ラン境界7210か、右ラン境界7310のいずれかにある。圧縮プロセス8000は、マーチング・レコード5180の値を割当て、共有頂点7700が左右いずれのラン境界にあるか識別する。三角ランが走査されると(7005)、三角ランのマーチング・エッジ7600はそれぞれ、これらの値の1つで識別され、これらの値のシーケンスは対応する三角ラン7000に関連付けられたマーチング・コマンドのシーケンス5190である。例えば三角ラン7000が走査されると(7005)、マーチング・エッジ7500、7600、7800が横切られる。マーチング・エッジ7500を横切ったとき、値”0”が与えられて、左ラン境界7210を進むことによって、三角形7505が構成されたことが示される。言い換えると、三角形7505の辺の1つは左ラン境界7210にある。マーチング・エッジ7600を横切ったときは値”1”が与えられて、右ラン境界7310を進むことによって三角形7605が構成されたことが示される。言い換えると、三角形7605の辺の1つは右ラン境界7310にある。次に、ラスト・マーチング・エッジ7800が横切られるときは値を割当てる必要はない(後述)。

【0063】図18は、単純三角メッシュを圧縮する方法8000のフローチャートである。先に述べた通り、単純三角メッシュを圧縮することは次のステップを含む。単純三角メッシュの頂点とエッジのグラフで頂点スパニング・ツリーと三角スパニング・ツリーを作成するステップ8100、単純三角メッシュの頂点位置に関連データ・レコードで記述するステップ8300、三角ラン・テーブルで三角スパニング・ツリーを記述するステップ8400、及びマーチング・レコードを確立するステップ8500である。圧縮プロセス8000でこれらのデータ構造が定義されると、データ構造はメモリに保存(8600)或いはまたネットワーク(インターネット)等の通信リンクで転送することができる(8600)。

【0064】ステップ8100で頂点スパニング・ツリーと三角スパニング・ツリーが作成される。これは単純三角メッシュ6000のどのエッジ2120がカット・エッジ6105(すなわち頂点スパニング・ツリー6100に属するもの)か、また単純三角メッシュ6000のどのエッジ2120がマーチング・エッジ7600(すなわち三角スパニング・ツリーに属するもの)かを決定することによって行われる。このステップを行う方法はいくつかある。図23、図24を参照されたい。

【0065】ステップ8200、8300、8400、

8500は、単純三角メッシュを圧縮された形で表すためにデータ構造5000を埋めるものである。

【0066】ステップ8200は、頂点ラン・テーブル5030にレコード5040を作成することによってデータ構造5000の頂点スパニング・ツリー6100を定義する。これについては図25で詳しく説明する。

【0067】ステップ8300は、単純三角メッシュの全ての頂点に対応した補正項を決定する。このステップの実行は任意である。ステップ8300は、ステップ8200で決定された走査順序を使い、与えられた各頂点の親元を識別する。補正項は次に、先に述べたように与えられた各頂点について決定され、データ構造5000のレコード5080に置かれる。このステップ8300で、色、法線、テキスチャ座標、その他の情報を表すデータは追加データ・レコード5090に置かれることに注意されたい。

【0068】ステップ8400は、三角ラン・テーブル5130にレコード5140を作成することによってデータ構造5000の三角スパニング・ツリー6690を定義する。これについては図26で詳しく説明している。

【0069】ステップ8500は、データ構造5000のマーキング・レコード5180を確立する。これについては図17で説明している。

【0070】ステップ8600は、データ構造5000をメモリに保存し、或いはまたデータ構造5000の情報を通信リンクで転送する。この情報は更に、これまでの手法により圧縮／暗号化できる。

【0071】図19乃至図22は、切断方法が異なると、ラン数の異なる頂点スパニング・ツリーと三角スパニング・ツリーが作成されることをビジュアルに示している。例えば単純三角メッシュ9000が、1つの頂点ラン9015で構成される頂点スパニング・ツリー9010のエッジで切断されるとき、三角スパニング・ツリー9200を3つの三角ラン(9210、9220、9230)で作成することができる。しかし、同じ単純三角メッシュ9100を、5つの頂点ラン(9115等)を持つ頂点スパニング・ツリー9110のエッジで切断すると、17の三角ラン(9310、9320等)を持つ三角スパニング・ツリー9300が得られる。

【0072】図23は、頂点スパニング・ツリーと三角スパニング・ツリーを作成する好適な実施例で行われるステップを示すフローチャートである。ステップ10100で、単純三角メッシュの各エッジにコストが割当てられる。ステップ10200では、最小スパニング・ツリー構成アルゴリズムを使用して、頂点スパニング・ツリーが総コストが最小のスパニング・ツリーとして構成される。従来の技術ではこのようなアルゴリズムは数多く知られている。エッジは最初にコストを上げることによって順序付けされる。ステップ10300で、頂点ス

パニング・ツリーによっては用いられなかった全てのエッジを使用して三角スパニング・ツリーが構成される。

【0073】より好適な実施例では、ステップ10100で、メッシュの各エッジに割当てられるコストはエッジの長さである。別のより好適な実施例の場合、メッシュの各エッジに割当てられるコストはエッジの中間点から頂点スパニング・ツリーの頂点ルート・ノードに関連付けられた頂点までのユークリッド距離である。こうして頂点スパニング・ツリーのルートに近いエッジは、遠いものより前にあるとみなされ、両方のツリーが頂点スパニング・ツリーのルートから伸びてメッシュ全体に及ぶ。

【0074】図24は、頂点スパニング・ツリーと三角スパニング・ツリーを作成するステップ8100の他の好適な実施例で行われるステップを示すフローチャートである。8100の例で単純三角メッシュは、三角化された層のシーケンスに圧縮解除された後、頂点スパニング・ツリーと三角スパニング・ツリーの両方が、これらの層のそれぞれをこの層シーケンスで走査することでインクリメンタルに構成される。このプロセスは直観的には、図21に示すように同心円状の輪を切り出し、らせん状につなげていくことでオレンジの皮をむくことに似ている。

【0075】最初、頂点スパニング・ツリー6100と三角スパニング・ツリー6690は両方ともエッジがない。ステップ11100で現在の頂点の集合が選択される。最初、現在の頂点の集合には頂点が1つしかない。ステップ11150は、単純三角メッシュの全ての三角形が走査されているかどうか確認する。その場合プロセス8100は停止する。そうでなければステップ11200で、ステップ11100で選択された現在の頂点の集合の頂点を含む未処理の(走査されていない)三角形が全て選択される。選択された11200の三角形の集合は、現在の三角化層と呼ばれる。こうした現在の三角化層の例は、図25の12100として示している。ステップ11300で、割当てられていない全てのエッジ(後述)と、現在の頂点の集合である頂点ノードの少なくとも1つを持つエッジが、層エッジの集合として選択される。これら層エッジ12200は、現在の三角化層の内側のエッジである。

【0076】ステップ11310、11320、11340、11350、11360は、層エッジを三角スパニング・ツリー6690か、頂点スパニング・ツリー6100に割当てるステップである。ステップ11310は最初、処理する層エッジが残っているかどうか確認する。そうでなければプロセス8100はステップ11400に進む。層エッジが残っていれば、与えられたエッジが選択され(ステップ11320)、ステップ11340で選択された層エッジが三角スパニング・ツリーに割当てられるか確認される。層エッジは三角スパニング

・ツリーでループを作らない場合にのみ三角スパニング・ツリーに割当てることができる。これは周知の手法で確認される。ステップ11350で層エッジは三角スパニング・ツリーでループを作らない場合に三角スパニング・ツリーに割当てられる。層エッジが三角スパニング・ツリーでループを作る場合は、ステップ11360で、層エッジが頂点スパニング・ツリーに割当てられる。単純三角メッシュでは、層エッジが三角スパニング・ツリーでループを作る場合は頂点スパニング・ツリーではループを作らないということに注意されたい。従って、この方法で層エッジを割当てること、頂点スパニング・ツリーか三角スパニング・ツリーのいずれかでループが作られない。

【0077】このプロセス8100の残りのステップは、残った割当てられていない三角化層の三角形のエッジを割当てて、これらのエッジは、現在の三角化層を三角化層のシーケンスの次の三角化層から分けるので分離エッジと呼ばれる。ステップ11400はこれらの分離エッジを選択する。

【0078】ステップ11410、11420、11440、11450、11460は、分離エッジを頂点スパニング・ツリー6100か三角スパニング・ツリー6690に割当ててステップである。ステップ11410は最初、処理する分離エッジが残っているか確認する。そうでなければプロセス8100はステップ11500に進む。分離エッジが残っていれば与えられたエッジが選択され（ステップ11420）、ステップ11440で選択された分離エッジを頂点スパニング・ツリーに割当てられるかどうか確認される。分離エッジは、頂点スパニング・ツリーでループを作らない場合にのみ割当てることができる。これは周知の手法により確認される。ステップ11450で分離エッジは頂点スパニング・ツリーでループを作らない場合は頂点スパニング・ツリーに割当てられる。分離エッジが頂点スパニング・ツリーでループを作る場合、ステップ11460で三角スパニング・ツリーに層エッジが割当てられる。単純三角メッシュでは、層エッジが頂点スパニング・ツリーでループを作る場合は、三角スパニング・ツリーではループを作らない。従って、この方法で分離エッジを割当てると、三角スパニング・ツリーか頂点スパニング・ツリーにループは作られない。

【0079】ステップ11500は、分離エッジに含まれる頂点の集合を、現在の頂点の集合としてセットし、プロセス8100はステップ11500に進む。

【0080】図26に示しているプロセス8200は、頂点ラン・テーブルにレコード5040を作成する。ステップ13005でプロセス8200により、頂点リーフ・ノード6301の1つが頂点ルート・ノードとして任意に選択される。選択された頂点リーフ・ノードは第1ラン・ノードにセットされる（13010）。（頂点

リーフ・ノード6301は先に定義されている。）これにより頂点ランの第1ノードが定義される。

【0081】ステップ13100は、先に第1ラン・ノードから走査されたランを選択する。（頂点リーフ・ノード6301から始めるとき、走査することができるランは1つしかない。）更にそのランは、頂点走査順序6380で選択される。単純三角メッシュ6000の向きはどのノードの周りでも、特に第1ラン・ノードの周りで、回転の方向（時計回りか反時計回り）を決定する。ある好適な実施例の場合、第1ラン・ノード周りの頂点走査順序6380は、第1ラン・ノードに入るエッジから始めて、第1ラン・ノードから離れて時計回りにランを取ることにによって決定される。これに代えて反時計回りにランを取ることもできる。ただし、単純三角化メッシュの全ての第1ラン・ノードについて同じ頂点走査順序6380を用いる必要がある。

【0082】ステップ13200は、処理されている現在のランが、第1ラン・ノードを共通に持つランの処理対象の最後のランかどうか確認される。この確認の結果、頂点ラン・テーブル5030の頂点ラン・レコード5040の“頂点ラスト”・フィールド5070の値の場所が決定される。ステップ13300は、ステップ13100で選択されたランのエッジをカウントし、この値をデータ構造5000の“頂点ラン長さ”フィールド5050に置く。

【0083】ステップ13400は、ランがリーフで終わるかどうか確認する。この確認の結果、頂点ラン・テーブル5030の頂点ラン・レコード5040の“頂点リーフ”・フィールド5060の値位置が決定される。ランがリーフで終わらない（すなわち分岐で終わる）場合、最初のラン・ノードが現在のランの最後のノード（分岐ノード）にセットされ、プロセス8200はステップ13100に戻る。

【0084】ランがリーフで終わる場合（13400）、プロセス8200は、単純三角メッシュの全てのランがプロセス8200で走査されているかどうか確認する（13500）。全てのランが走査されている場合、プロセス8200は終了する。そうでなければステップ13600が実行される。

【0085】ステップ13600は最初のラン・ノードを走査されていないランを持つ選択済みリーフ・ノードの前の（親の）最後に走査された分岐ノードにセットし、ステップ13100からまたプロセスを開始する。従って、プロセス8200は、頂点リーフ・ノードから始まって、頂点スパニング・ツリーの全てのランが識別され、データ構造5000の頂点ラン・レコード5040の値が割当てられるまで、頂点スパニング・ツリーを上がっていく。

【0086】図27のプロセス8400は、三角ランのテーブルにレコード5140を作成する。ステップ14

005でプロセス8400は任意、三角リーフ・ノード6610の1つを三角ルート・ノードとして選択する。選択された三角リーフ・ノードは第1三角ラン・ノードにセットされる(14010)。(三角リーフ・ノード6610は先に定義されている。)これにより三角ランの最初のノードが定義される。

【0087】ステップ14100は、先に、最初の三角ラン・ノードから始めて走査されなかったランを選択する。(三角リーフ・ノード6610から始めたとき、走査することのできるランは1つしかない。)更にランは三角形走査順序6680で選択される。単純三角メッシュ6000の向きは全てのノード、特に第1三角ラン・ノードの周りの回転方向(時計回りか反時計回り)を決定する。ある好適な実施例の場合、第1三角ラン・ノード周りの三角形走査順序6680は、第1三角ラン・ノードに入るエッジから始めて、第1三角ラン・ノードから離れる三角ランを時計回りに取ることによって決定される。また反時計回りに三角ランを取ることもできる。ただし単純三角化メッシュの全ての第1三角ラン・ノードについて同じ三角形走査順序6680を使用しなければならない。更に三角形走査順序6680は、頂点走査順序6380と同じ方向で取る必要がある。

【0088】ステップ14200は、処理されている現在の三角ランが、第1三角ラン・ノードを共通に持つランの処理対象の最後の三角ランであるかどうかを確認する。ステップ14300は、ステップ14100で選択された三角ランのマーチング・エッジ7600をカウントし、この値をデータ構造500の"三角ラン長さ"フィールド5150に置く。

【0089】ステップ14400は三角ランがリーフで終わるかどうかを確認する。このテストの結果により、三角ラン・テーブル5130の三角ラン・レコード5140の"三角形リーフ"・フィールド5160の値位置が決定される。三角ランがリーフで終わらない(すなわち分岐で終わる)場合、最初の三角ラン・ノードが、現在のランの最後の三角ノードにセットされ(分岐ノード)、プロセス8400はステップ14100に戻る。

【0090】ランがリーフで終わると(14400)、プロセス8400により、単純三角メッシュの全てのランがプロセス8400で走査されているかどうかを確認される(ステップ14500)。全てのランが走査されている場合、プロセス8400は終了する。でなければステップ14600が実行される。

【0091】ステップ14600は、第1ラン・ノードを、走査されていないランを持つ選択済みリーフ・ノードの前の(親である)最後に走査された分岐ノードにセットし、再びプロセスを開始する(ステップ14100)。従って、プロセス8400は三角リーフ・ノードから始まり、三角スパニング・ツリーの全てのランが識別され、データ構造5000の三角ラン・レコード51

40の値が割当てられるまで、三角スパニング・ツリーが走査される。

【0092】図28は、好適な圧縮解除プロセス15000のフローチャートである。

【0093】単純三角メッシュの圧縮解除プロセス15000は次のステップを伴う。データ構造5000のアクセス/受信のステップ15100、頂点位置テーブルの再構成のステップ15200、境界ループ・ノードと頂点スパニング・ツリー・ノードの対応をとるルックアップ・テーブルを作成するステップ15300、使用順序でのY境界ループ・ノードの相対索引を計算するステップ15400、及び三角ストリップを再構成しリンクするステップ15500である。

【0094】また、圧縮された表現にデータ・レコード5090が追加され、色、法線、テキスチャ座標等が記述される場合、色、法線、テキスチャ座標等のテーブルが、頂点位置のテーブルが構成されるのと同じように構成される。

【0095】ステップ15100でデータ構造5000がメモリからアクセスされるか、通信リンクを通して受信される。後述するように、このデータ構造5000の情報は更に圧縮、或いは暗号化できる。

【0096】ステップ15200は、頂点ラン・テーブル5030の頂点スパニング・ツリー情報の構造及び関連するデータ・レコード5080から、頂点位置アレイを構成する。同時に頂点走査順序6380に従って頂点スパニング・ツリーのノードを走査する。好適な実施例の場合、ツリー走査時に頂点ノードが走査されるとき、頂点の親元に対する索引のアレイが維持され、予測頂点位置に補正項を追加することによって頂点位置が計算される。

【0097】ステップ15300は、境界ループ・ノードと頂点スパニング・ツリー・ノードの対応をとるルックアップ・テーブルを構成する。一例として、境界ループ・ノード6401(図11乃至図16)は、境界ループ・ノード6402、6403として頂点スパニング・ツリー・ノード6301に関連付けられる。境界ループ・ノード6402、6403は両方とも頂点スパニング・ツリー・ノード6302に関連付けられる。従って、頂点スパニング・ツリー6300がステップ15200で走査されている間、頂点スパニング・ツリー・ノードそれぞれの境界ループ・ノード情報が確認され、ルックアップ・テーブルに入力される。

【0098】境界ループは、頂点スパニング・ツリーの再帰走査中に構成され、ルックアップ・テーブルによって表される。ツリーを降りながら検出された頂点への参照は、走査時にテーブルに追加される。リーフ頂点を除いてこれらの参照はスタックにもプッシュされる。頂点ランがどう連結されて頂点スパニング・ツリーを形成するかを示す各頂点ランのラストとリーフの各フィールド

は、ツリー走査とスタックのポッピングを制御するために用いられる。

【0099】ステップ15400は、三角スパニング・ツリーの走査順序6680でのY境界ループ・ノードの相対索引を計算する。このステップは、三角スパニング・ツリーの分岐ノードで三角形を作成する方法を決定するために、境界ループ・ノードの三角形の第3の頂点を見つける。“左”と“右”の頂点はすでに求められていることに注意されたい。図29参照。

【0100】図29はY境界ループ・ノードを示す。三角ラン16100には第1ノード16200と最終ノード16300がある。三角ランの最後のマーチング・エッジ16400は、最終ノード16300に関連付けられた最後の三角形を三角ランの前の三角形につなぐ。最後のマーチング・エッジ16400はまた、最後の左境界ループ・ノード16500を最後の右境界ループ・ノード16600につなぐ。最後の三角形は左右の最後の境界ループ・ノードを第3境界ノード16700につなぐ。第3境界ノード16700のことをここではY境界ループ・ノードと呼んでいる。最後の三角形を構成するには、第3境界ノード16700は最後の三角形、つまり分岐ノード側の三角形の全てについて前処理の段階で決定しなければならない。この情報は三角ランのテーブル5130に暗示的に保持される。従って、三角ランの三角形を構成し始める前に、最後の三角形に関するこの情報を明示的にしなければならない。

【0101】そのため、境界ループ周りの左右いずれかの方向から、各分岐三角形からの距離が決定される。各分岐三角形について、左右いずれかの頂点からY境界ループ・ノードへのループに沿った距離、左分岐境界長と右分岐境界長が再帰により計算できる。長さ n のランから始まる分岐の境界の長さは、 $n + n_L + n_R - 1$ に等しく、ここで n_L と n_R はランがリーフ三角形で終わる場合は1に等しく、ランが分岐三角形で終わる場合は分岐三角形の左右の分岐境界長に等しい。

【0102】分岐境界長は圧縮解除アルゴリズムの前処理段階として、各分岐について計算され、分岐境界長テーブルに保存される。三角形再構成段階で分岐三角形が検出されると、左分岐境界長を左頂点のループ索引に加算することによって、対応する分岐頂点のIDを確認することができる。境界ループ・テーブルは円形なので、この加算は境界ループの長さを法として行われる。

【0103】ステップ15500で、三角化多角形の三角形は3つの境界ループ・ノードのグループとして構成される。そのため三角ランのテーブル5130とマーチング・レコード5180が用いられる。マーチング・レコード5180の三角ルート・フィールド5185は、三角スパニング・ツリー6690の三角ルート・ノード6610に関連付けられた三角形の1つを構成する3つの連続した境界ループ・ノードを識別する。図17で説

明している通り、各三角ランの三角形の残りは、マーチング・コマンドに従って、ランの左側か右側を進むことによって再構成される。

【0104】これは好適な実施例の場合、いずれも最初は空である左頂点スタックと右頂点スタックの2つのスタックを使用した再帰により行われる。三角ランのテーブルの三角形ラン・レコードに記述される通り、現在の左右の頂点が n 個の三角形の三角ランの初めにあるとき、マーチング・パターンの次の n ビットはストリップの各側で走査されたルートの頂点数を識別する。

【0105】三角ランの三角形最終ノードが三角形分岐ノードなら、対応するY境界ループ・ノードに関連付けられた境界ループ・ノードは先に計算され、分岐境界長テーブルに保存された情報から決定され、現在の左右頂点とY境界ループ・ノードで決定される三角形が再構成され、Y境界ループ・ノードと現在の右頂点はそれぞれ、左右の頂点スタックにプッシュされ、現在の右頂点はY境界ループ・ノードに等しくセットされる。三角ランの三角形最終ノードが三角形リーフ・ノードのとき、現在の右頂点の後継（者）は、現在の左頂点の先祖に等しくする必要がある。その場合、リーフ三角形が再構成され、現在の左右頂点が可能なら左右頂点スタックからポップ・オフされる。スタックが空なら三角形は全て再構成されているので再構成プロセスは停止する。

【0106】好適な実施例の場合、法線、色及びテキスト座標は量子化され、予測誤差は、頂点位置の場合のようにエントロピ・エンコードされる。他の好適な実施例では、法線はDeeringの非直線法により量子化される。

【0107】図30は、本発明の好適な実施例が適用されるコンピュータ・システムの例17300のブロック図である。この好適な実施例には、1つ以上のアプリケーション・プログラム17302が含まれる。アプリケーション・プログラム17302の一種として、オブティマイザ17306を含むコンパイラ17305がある。コンパイラ17305とオブティマイザ17306はソース（他のアプリケーション・プログラム17302等）・プログラムを、最適化された実行可能コードに変換するよう設定される。一般的にソース・プログラムは、最適化形式に変換されてから実行可能コードに変換される。コンパイラ17305とオブティマイザ17306は、コンピュータ・プラットフォーム17304で動作する。プラットフォーム17304はハードウェア装置17312を含む。システム17300で実行されるアプリケーション・プログラム17302の一部は、先に述べた圧縮プロセス8000と圧縮解除プロセス15000を含む。

【0108】ハードウェア装置17312は、CPU（中央処理装置）17316、RAM（ランダム・アクセス・メモリ）17314、及び入力/出力インターフ

ェイス17318を含む。マイクロ命令コード17310、例えば縮小命令セット、プラットフォーム17304に追加できる。グラフィカル・インターフェイスまたは端末17326、データ記憶装置17330、印刷装置17334等、様々な周辺装置をコンピュータ・プラットフォーム17304に接続できる。オペレーティング・システム17308は、コンピュータ・システム17300の様々な構成要素の動作を調整する。このようなコンピュータ・システム17300の一例として、IBM RISC System/6000がある(RISC System/6000はIBM社の商標である)。言うまでもなく当業者には、これと同等な多くのコンピュータ・システム17300に精通しているはずである。

【0109】図31は、通信リンク18220を通して第2コンピュータ18210に接続された第1コンピュータ18200の図である。通信リンクの例として、シリアル・リンク(RS-232)、パラレル・リンク、無線周波数リンク、赤外線リンクがある。ネットワーク(LAN、WAN)も従来の通信リンクの例である。ネットワークは一般に普及している。ネットワークの一例としてインターネットがある。詳しくは、Attanasioらによる1992年10月14日付け、米国特許番号第5371852号を参照されたい。コンピュータ18200は三角メッシュを圧縮するため、幾何圧縮プロセス8000を実行し、得られたデータ構造5000を通信リンク18220を通して送る。コンピュータ18210はデータ構造5000を受信し、三角メッシュを圧縮解除するため幾何圧縮解除プロセス15000を実行する。

【0110】幾何モデルを圧縮/圧縮解除する一般的方法については、米国特許出願番号第010045号、"Compression of Geometric Models Using Spanning Trees"を参照されたい。

【0111】当業者には、本発明の開示内容により、本発明の他の実施例も明らかになろう。そうした実施例も本発明の適用範囲にある。

【0112】まとめとして、本発明の構成に関して以下の事項を開示する。

【0113】(1) n 次元空間の点の集合を表すコンピュータ・メモリのデータ構造であって、ノードを持ち、各ノードが正規ノード、分岐ノード、またはリーフ・ノードであるルート付きツリーを記述し、前記ルート付きツリーの各ノードは、 n 次元空間の点を表し、またレコードを持ち、各レコードは前記ルート付きツリーのラン、すなわちリーフ・ノードか分岐ノードである第1ノードと、リーフ・ノードか分岐ノードである最終ノードを持つランを表し、該ランは、前記第1ノードと前記最終ノードの間の正規ノードを連結し、前記ランの連続したノードのペアそれぞれの連結部はエッジであり、各レコードは、前記ランのエッジ数で表されるラン長フィー

ルド、前記ランが前記リーフ・ノードで終わるかどうかが示すリーフ・フィールド、及び前記レコードによって表されるランが、前記ルート付きツリーと同じ前記第1ノードを持つ最後のものかどうかを示すラスト・ラン・フィールドを持ち、前記レコードは、ルート・ノードに対してツリー走査順序で与えられる、前記ランのテーブル、を含む、データ構造。

(2) 関連データ・フィールドを含み、各関連データ・フィールドは補正項であり、各補正項は、現在の点の実際の位置と現在の点の予測位置の差であり、現在の点は、カレント・ノードと呼ばれるノードの1つと関連付けられた、前記(1)記載のデータ構造。

(3) 前記現在の点の予測位置は、現在の点の親元点を使用する予測法により決定され、該親元点は、前記カレント・ノードから前記ルート・ノードへの固有パスにあるノードに対応する、前記(2)記載のデータ構造。

(4) 前記関連データ・フィールドは固定小数点形式である、前記(1)記載のデータ構造。

(5) 前記関連データ・レコードは圧縮される、前記(1)記載のデータ構造。

(6) 前記関連データ・レコードは、ハフマン・エンコーディング、算術コーディング、シャノン・ファノ・エリヤス・コーディング、レンベル・ジブ・コーディングのいずれか1つにより圧縮される、前記(5)記載のデータ構造。

(7) 点位置をコンピュータの記憶装置に保存するために用いられる、前記(1)記載のデータ構造。

(8) コンピュータから通信リンクを通して送られる、前記(1)記載のデータ構造。

(9) 通信リンクを通してコンピュータにより受信される、前記(1)記載のデータ構造。

(10) 前記点は、 n 次元幾何形状の頂点を表し、各ツリー・エッジは、前記幾何形状の形状エッジに対応する、前記(1)記載のデータ構造。

(11) n 次元空間で単純三角メッシュを表すデータ構造であって、頂点ノードを持ち、各頂点ノードが正規頂点ノード、分岐頂点ノード、またはリーフ頂点ノードであり、前記単純三角メッシュの頂点に関連付けられた頂点スパニング・ツリーを記述する頂点ランのテーブルであって、頂点ラン・レコードを持ち、各頂点ラン・レコードは前記頂点スパニング・ツリーの頂点ランを表し、リーフ頂点ノードか分岐頂点ノードである第1頂点ノードと、リーフ頂点ノードか分岐頂点ノードであるラスト頂点ノードとを持ち、前記第1頂点ノードとラスト頂点ノードとの間の正規頂点ノードを連結し、そこで、連続した頂点ノードのペアそれぞれの間の連結部は頂点ツリー・エッジであり、各頂点ラン・レコードは頂点ランの頂点ツリー・エッジ数により与えられる頂点ラン・フィールドの長さ、頂点ラン・レコードにより表される頂点ランのラスト頂点ノードがリーフ頂点ノードかどうかを

示す頂点リーフ・フィールド、及び頂点ランが、同じ前記第1頂点ノードを共有するラスト頂点ランかどうかを示す頂点ラスト・ラン・フィールドを持ち、前記頂点ランは、頂点ルート・ノードに対する前記頂点スパニング・ツリーの走査順序により与えられる、前記頂点ランのテーブルと、三角ノードを持ち、各三角ノードは、正規三角ノード、分岐三角ノード、またはリーフ三角ノードであり、前記単純メッシュの三角形に関連付けられた三角スパニング・ツリーを記述する、三角ランのテーブルであって、三角ラン・レコードを持ち、各三角ラン・レコードは前記三角スパニング・ツリーの三角ランを表し、前記三角ランは、リーフ三角ノードか分岐三角ノードである第1三角ノードと、リーフ三角ノードか分岐三角ノードであるラスト三角ノードとを持ち、前記第1三角ノードと前記ラスト三角ノードとの間の正規三角ノードを連結し、そこで、連続した三角ノードのペアそれぞれの間の連結部は三角ツリー・エッジであり、各三角ラン・レコードは、前記三角ラン・レコードで表される三角ランの三角ツリー・エッジ数により与えられる三角ラン長フィールドと、前記三角レコードで表される三角ランの前記ラスト三角ノードがリーフ三角ノードかどうかを示す三角リーフ・フィールドとを持ち、前記三角ランは、三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序により与えられる三角ランのテーブルと、三角ルート・フィールドと、マーチング・コマンドのシーケンスとを持ち、前記三角ルート・フィールドは、前記三角スパニング・ツリーの三角ルート・ノードに関連付けられた三角形を構成する方法を記述し、マーチング・コマンドの各シーケンスは、前記三角ランの1つから三角形を構成する方法を示すため、前記三角ランの左ラン境界または右ラン境界に沿った次の頂点に進み、前記マーチング・コマンドは、前記三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序により与えられるマーチング・レコードと、関連データ・フィールドを持ち、各関連データ・フィールドは、前記単純三角メッシュの頂点の1つの位置に関する情報を持ち、前記関連データ・フィールドは、前記頂点ルート・ノードに対する前記頂点スパニング・ツリーの走査順序により与えられる、関連データ・レコードと、を含む、データ構造。

(12) 前記頂点スパニング・ツリーの走査順序は、深さ優先、幅優先、前順、後順のいずれか1つを含む、前記(11)記載のデータ構造。

(13) 各関連データ・フィールドは補正項であり、各補正項は、現在の頂点の実際の位置と現在の頂点の予測位置の差であり、現在の頂点は、カレント・ノードと呼ばれるノードの1つと関連付けられた、前記(11)記載のデータ構造。

(14) 前記現在の頂点の予測位置は、現在の頂点の親元の頂点を使用する予測法により決定され、親元頂点

は、現在の頂点ノードから頂点ルート・ノードへの固有パスにあるノードに対応する、前記(13)記載のデータ構造。

(15) 前記関連データ・フィールドは固定小数点形式である。前記(11)記載のデータ構造。

(16) 前記関連データ・レコードは圧縮される、前記(11)記載のデータ構造。

(17) 前記関連データ・レコードは、ハフマン・エンコーディング、算術コーディング、シャノン・ファノ・エリヤス・コーディング、レンペル・ジブ・コーディングのいずれか1つにより圧縮される、前記(16)記載のデータ構造。

(18) 前記三角スパニング・ツリーの走査順序は、前記三角ラン・テーブルの三角ラン・レコードの順序により決定される、前記(11)記載のデータ構造。

(19) 前記三角スパニング・ツリーの走査順序は、深さ優先、幅優先、前順、後順のいずれか1つを含む、前記(11)記載のデータ構造。

(20) 前記マーチング・レコードは圧縮される、前記(11)記載のデータ構造。

(21) 前記マーチング・レコードは、ハフマン・エンコーディング、算術コーディング、シャノン・ファノ・エリヤス・コーディング、レンペル・ジブ・コーディングのいずれか1つにより圧縮される、前記(20)記載のデータ構造。

(22) 前記マーチング・コマンドはそれぞれビットである、前記(11)記載のデータ構造。

(23) 追加データ・レコードを含み、追加される各関連データ・レコードは、追加データ・フィールドを持ち、前記追加データ・フィールドはそれぞれ追加情報を持つ、前記(11)記載のデータ構造。

(24) 前記追加情報は、法線、色及びテキスト座標を含む、前記(23)記載のデータ構造。

(25) 前記単純三角メッシュをコンピュータの記憶装置に保存するために用いられる、前記(11)記載のデータ構造。

(26) コンピュータから通信リンクを通して送られる、前記(11)記載のデータ構造。

(27) 通信リンクを通してコンピュータにより受信される、前記(11)記載のデータ構造。

(28) 複数の連結成分を持つ n 次元空間で前記単純三角メッシュの連結成分を表す、前記(11)記載のデータ構造。

(29) 幾何モデルを圧縮し圧縮解除する1つ以上のコンピュータ・システムであって、各コンピュータ・システムは、プロセスを実行する中央処理装置と、メモリと、前記メモリに存在し、それぞれが正規頂点ノード、分岐頂点ノード、またはリーフ頂点ノードであり、メッシュの頂点に関連付けられた頂点ノードを持つ頂点スパニング・ツリーを記述する頂点ランのテーブルであって

て、頂点ラン・レコードを持ち、各頂点ラン・レコードは、前記頂点スパニング・ツリーの頂点ランを表し、前記頂点ランはリーフ頂点ノードか分岐頂点ノードである第1頂点ノードと、リーフ頂点ノードか分岐頂点ノードであるラスト頂点ノードとを持ち、前記第1頂点ノードと前記ラスト頂点ノードとの間の正規頂点ノードを連結し、そこで、連続した頂点ノードのペアそれぞれの間の連結部は、頂点ツリー・エッジであり、各頂点ラン・レコードは、前記頂点ランの頂点ツリー・エッジ数で与えられる頂点ラン長フィールド、前記頂点ラン・レコードで表される頂点ランの前記ラスト頂点ノードがリーフ頂点ノードかどうかを示す頂点リーフ・フィールド、及び前記頂点ランが、同じ前記第1頂点ノードを共有する前記ラスト頂点ランであるかどうかを示す頂点ラスト・ラン・フィールドを持ち、前記頂点ランは、頂点ルート・ノードに対する前記頂点スパニング・ツリーの走査順序により与えられる、頂点ラン・テーブルと、前記メモリに存在し、それぞれが正規三角ノード、分岐三角ノード、またはリーフ三角ノードであり、前記単純三角メッシュの三角形に関連付けられた三角ノードを持つ三角スパニング・ツリーを記述する、三角ランのテーブルであって、三角ラン・レコードを持ち、各三角ラン・レコードは前記三角スパニング・ツリーの三角ランを表し、前記三角ランはリーフ三角ノードか分岐三角ノードである第1三角ノードと、リーフ三角ノードか分岐三角ノードであるラスト三角ノードとを持ち、前記ランは、前記第1三角ノードと前記ラスト三角ノードの間の正規三角ノードを連結し、そこで、連続した三角ノードのペアそれぞれの間の連結部は、三角ツリー・エッジであり、各三角ラン・レコードは、前記三角ラン・レコードで表される三角ランの三角ツリー・エッジ数で与えられる三角ラン長フィールドと、前記三角レコードで表される三角ランのラスト三角ノードが、リーフ三角ノードかどうかを示す三角リーフ・フィールドを持ち、前記三角ランは、三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序により与えられる、三角ラン・テーブルと、前記メモリに存在し、三角ルート・フィールドと、マーチング・コマンドのシーケンスを持ち、前記三角ルート・フィールドは、前記三角スパニング・ツリーの三角ルート・ノードに関連付けられた三角形を構成する方法を記述し、マーチング・コマンドの各シーケンスは、前記三角ランの1つから三角形を構成する方法を示すために、前記三角ランの左ラン境界または右ラン境界に沿った次の頂点に進み、前記マーチング・コマンドは、前記三角ルート・ノードに対する前記三角スパニング・ツリーの走査順序で与えられる、マーチング・レコードと、を含む、システム。

(30) 2つ以上のコンピュータ・システムが通信リンクで接続された、前記(29)記載のシステム。

(31) 前記通信リンクは、インターネット、イントラ

ネット、広域ネットワーク、ローカル・エリア・ネットワーク、無線周波数リンク、赤外線リンク及びシリアル通信リンクのいずれかを含む、前記(30)記載のシステム。

(32) 前記頂点ラン・テーブル、三角ラン・テーブル及びマーチング・レコードは、第1コンピュータ・システムから、前記通信リンクで接続された第2コンピュータ・システムへ転送される、前記(30)記載のシステム。

(33) 前記頂点ラン・テーブル、三角ラン・テーブル及びマーチング・レコードを、前記幾何モデルから作成する圧縮プロセスを含む、前記(29)記載のシステム。

(34) 前記頂点ラン・テーブル、三角ラン・テーブル及びマーチング・レコードから前記幾何モデルを作成する圧縮解除プロセスを含む、前記(29)記載の装置。

(35) 頂点スパニング・ツリーと三角スパニング・ツリーを幾何モデルから作成するステップと、前記頂点スパニング・ツリーを頂点ランのテーブルで記述するステップと、前記三角スパニング・ツリーを三角ランのテーブルで記述するステップと、前記頂点ラン・テーブルに従って前記三角スパニング・ツリーを走査しながら、前記三角スパニング・ツリーで三角形を構成する方法を定義するマーチング・レコードを設定するステップと、を含む、圧縮方法。

(36) 前記頂点スパニング・ツリーと前記三角スパニング・ツリーは、前記幾何モデルのエッジにコストを割当てるステップと、前記頂点スパニング・ツリーを総コストが最小のスパニング・ツリーと定義するステップと、前記三角スパニング・ツリーを前記頂点スパニング・ツリーに含まれるエッジで作成するステップと、で作成される、前記(35)記載の圧縮方法。

(37) 前記頂点スパニング・ツリーと前記三角スパニング・ツリーは、

- 前記幾何モデルの現在の頂点の集合を選択するステップと、
- 現在の頂点を含み、三角化された現在の層として走査されていない全ての三角形を選択するステップと、
- 選択されないエッジは分離エッジであり、層エッジとして割当てられておらず、前記現在の頂点を少なくとも1つ含む、全てのエッジを選択するステップと、
- 前記層エッジが、前記三角スパニング・ツリーにループを作らない場合には、前記層エッジそれぞれを前記三角スパニング・ツリーに割当て、そうでない場合は前記頂点スパニング・ツリーに割当てるステップと、
- 前記層エッジが前記三角スパニング・ツリーにループを作らない場合は、前記分離エッジそれぞれを前記三角スパニング・ツリーに割当て、そうでない場合は前記頂点スパニング・ツリーに割当てるステップと、
- 前記分離エッジを現在の頂点の集合として選択する

ステップと、

g. エッジがなくなるまでステップa乃至fを繰り返すステップと、

h. 割当てられた対応する前記エッジを使用して前記頂点スパニング・ツリーと前記三角スパニング・ツリーを作成するステップと、を含む、前記(35)記載の圧縮方法。

(38) 頂点位置を再構成するステップと、境界ループを構成するステップと、Y境界ループ・ノードの相対索引を決定するステップと、三角ストリップを再構成しリンクするステップと、を含む、圧縮解除方法。

【図面の簡単な説明】

【図1】従来の技術のツリー例を示す図である。

【図2】従来の技術のツリー例を示す図である。

【図3】従来の技術の三角メッシュ例を示す図である。

【図4】従来の技術の三角メッシュ例を示す図である。

【図5】従来の技術の三角メッシュ例を示す図である。

【図6】従来の技術の三角メッシュ例を示す図である。

【図7】従来の技術の三角メッシュ例を示す図である。

【図8】ツリーを表すデータ構造のブロック図である。

【図9】n次元空間の点の集合(集合の各点はルート付きツリーのノードに関連付けられる)を表すデータ構造のブロック図である。

【図10】本発明で開示した、n次元空間の単純三角メッシュを表すデータ構造のブロック図である。

【図11】三角メッシュの成分と図10のデータ構造の関係を示す図である。

【図12】三角メッシュの成分と図10のデータ構造の関係を示す図である。

【図13】三角メッシュの成分と図10のデータ構造の関係を示す図である。

【図14】三角メッシュの成分と図10のデータ構造の関係を示す図である。

【図15】三角メッシュの成分と図10のデータ構造の関係を示す図である。

【図16】三角メッシュの成分と図10のデータ構造の関係を示す図である。

【図17】マーチング・コマンドのシーケンスを持つマーチング・レコードの構造を示す図である。

【図18】単純三角メッシュを圧縮する方法のフローチャートを示す図である。

【図19】切断方法の違いにより、ラン数の異なる頂点スパニング・ツリーと三角スパニング・ツリーが作られることを示す図である。

【図20】切断方法の違いにより、ラン数の異なる頂点スパニング・ツリーと三角スパニング・ツリーが作られることを示す図である。

【図21】切断方法の違いにより、ラン数の異なる頂点スパニング・ツリーと三角スパニング・ツリーが作られることを示す図である。

【図22】切断方法の違いにより、ラン数の異なる頂点スパニング・ツリーと三角スパニング・ツリーが作られることを示す図である。

【図23】頂点スパニング・ツリーと三角スパニング・ツリーを作成する好適な実施例のステップのフローチャートを示す図である。

【図24】頂点スパニング・ツリーと三角スパニング・ツリーを作成する好適な実施例のステップのフローチャートを示す図である。

【図25】三角形の階層を示す図である。

【図26】頂点ランのテーブルに値を入力するフローチャートを示す図である。

【図27】三角ランのテーブルに値を入力するフローチャートを示す図である。

【図28】好適な圧縮解除プロセスのステップのフローチャートを示す図である。

【図29】Y境界ループ・ノード、すなわち三角ツリーの分岐点で三角形を作成する方法を示す図である。

【図30】本発明の好適な実施例が用いられるコンピュータ装置を示すブロック図である。

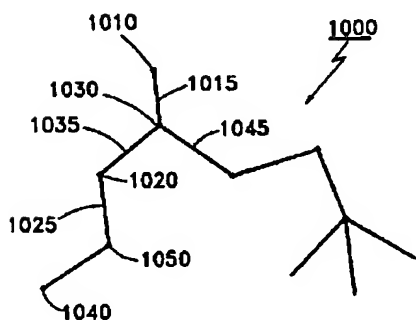
【図31】通信リンクを通して第2コンピュータに接続された第1コンピュータを示すブロック図である。

【符号の説明】

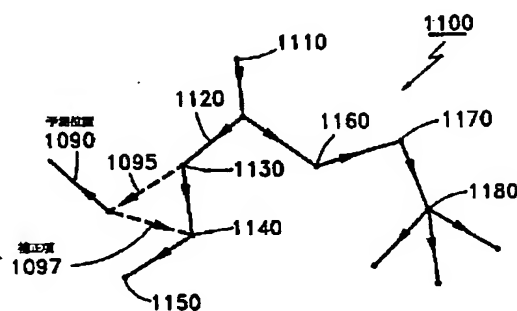
1000 ツリー
1015、1025、1035、1045、2120、
6105 エッジ
1010 リーフ・ノード
1020 正規ノード
1030 分岐ノード
1050、1140、1150、1180 ノード
1090 予測位置
1097 補正項
1100 ルート付きツリー
1110 ルート・ノード
1120 親元ノード
1130 点
2100、2200、2300、2400、2500
三角メッシュ
2110、2140、2150、2160、2165、
2170、2180、2230、2240、2250、
2410 頂点
2120、2340、2350 エッジ
2130、2320、2330 三角形
3000、4000、5000 データ構造
3100、5080 関連データ・レコード
3200 ランのテーブル
3300 レコード
3400 ラン長フィールド
3500 リーフ・フィールド
3600 ラスト・フィールド

- | | |
|---------------------------|---------------------------|
| 4200 データ・レコード | 6680 三角形走査順序 |
| 4300 データ・フィールド | 6690、9300 三角スパニング・ツリー |
| 5030、5040、6310、9015、9115 | 7000 三角ランの例 |
| 頂点ラン | 7005 ランの走査方向 |
| 5040 頂点ラン・レコード | 7210 左ラン境界 |
| 5050 頂点ラン長フィールド | 7310 右ラン境界 |
| 5060 頂点リーフ・フィールド | 7500、7600、16400 マーチング・エッジ |
| 5070 頂点ラスト・フィールド | 7505、7605 三角形 |
| 5090 追加データ・レコード | 7700 共有頂点 |
| 5095 追加データ・フィールド | 7800 ラスト・マーチング・エッジ |
| 5130、9210、9220、9230、9310、 | 8000 圧縮プロセス |
| 9320、16100 三角形ラン | 15000 圧縮解除プロセス |
| 5140 三角形ラン・レコード | 16200 第1ノード |
| 5150 三角形ラン長フィールド | 16300 最終ノード |
| 5160 三角形リーフ・フィールド | 16500 左境界ループ・ノード |
| 5180 マーチング・レコード | 16600 右境界ループ・ノード |
| 5185 三角ルート・フィールド | 16700 第3境界ノード |
| 5190 マーチング・コマンド | 17300 コンピュータ・システム |
| 6000、9000、9100 単純三角メッシュ | 17302 アプリケーション・プログラム |
| 6100、6300、9110、9200 頂点スパニ | 17304 コンピュータ・プラットフォーム |
| ング・ツリー | 17305 コンパイラ |
| 6105 頂点ツリー・エッジ | 17306 オプティマイザ |
| 6200、6400 境界ループ | 17308 オペレーティング・システム |
| 6205、6215 境界ループ・エッジ | 17312 ハードウェア装置 |
| 6301 頂点リーフ・ノード | 17314 RAM (ランダム・アクセス・メモリ) |
| 6302 頂点正規ノード | 17316 CPU (中央処理装置) |
| 6304 頂点分岐ノード | 17318 入力/出力インターフェイス |
| 6380 頂点走査順序 | 17310 マイクロ命令コード |
| 6401、6402、6403、6405、6406、 | 17326 グラフィカル・インターフェイスまたは端 |
| 6407、6408、6504、6514、6524 | 末 |
| 境界ループ・ノード | 17330 データ記憶装置 |
| 6500 三角化多角形 | 17334 印刷装置 |
| 6501、6502、6503、6504、6505、 | 18200 第1コンピュータ |
| 6506、6507、6508 多角形頂点 | 18210 第2コンピュータ |
| 6610、6620 三角ルート・ノード | 18220 通信リンク |
| 6611、6612 多角形三角形 | |

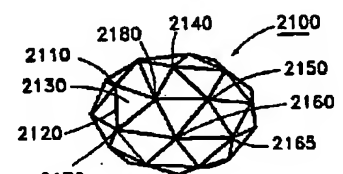
【図1】



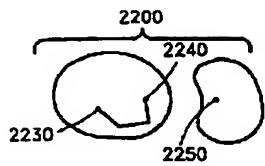
【図2】



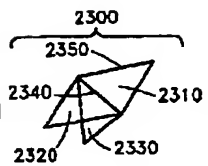
【図3】



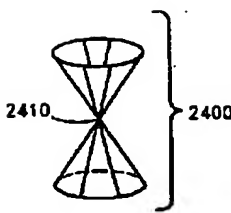
【図4】



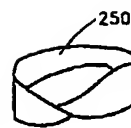
【図5】



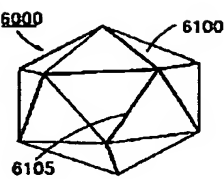
【図6】



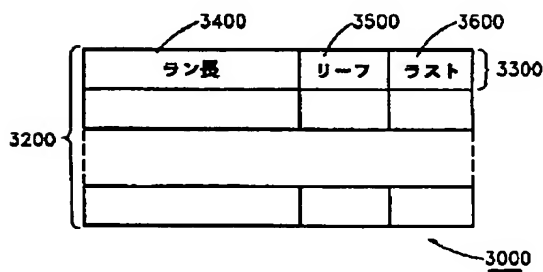
【図7】



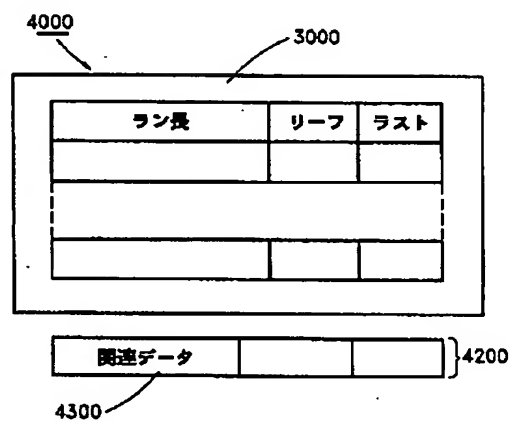
【図11】



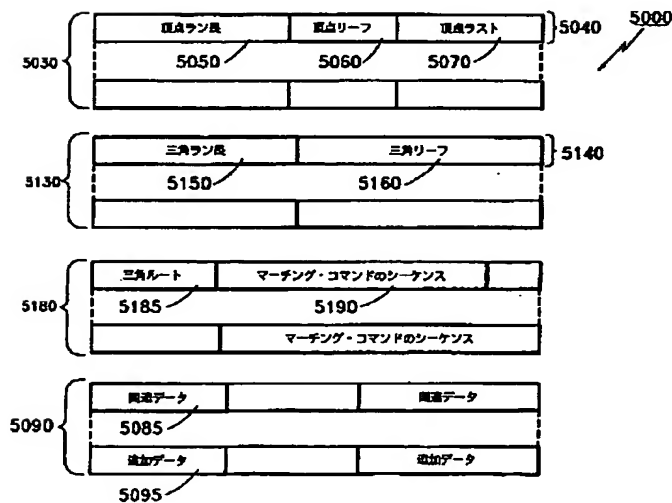
【図8】



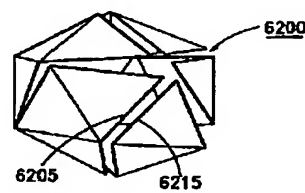
【図9】



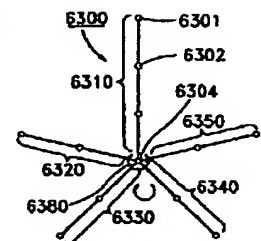
【図10】



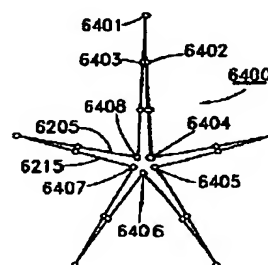
【図12】



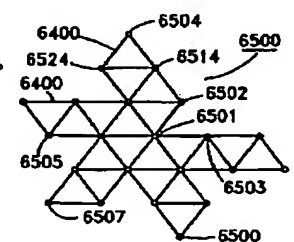
【図13】



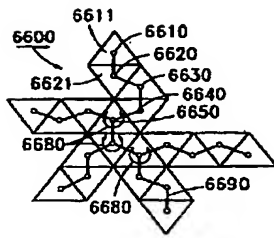
【図14】



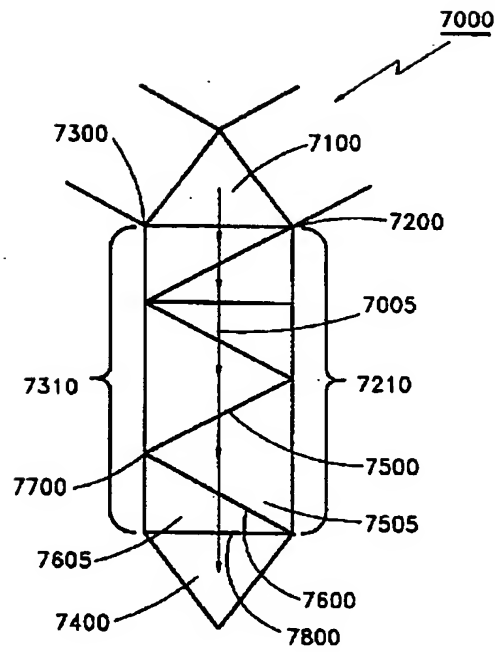
【図15】



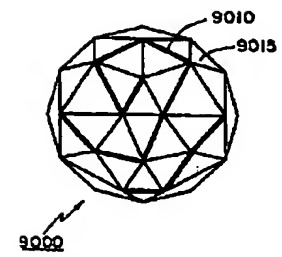
【図16】



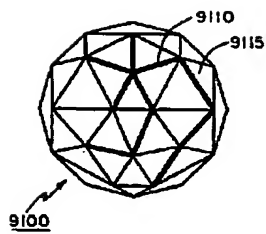
【図17】



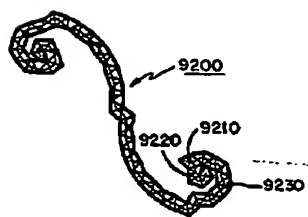
【図19】



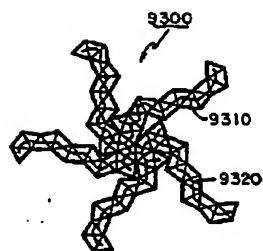
【図20】



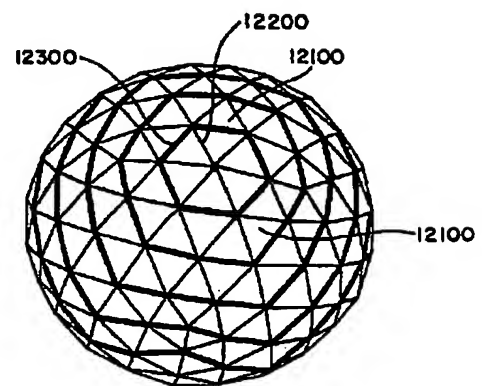
【図21】



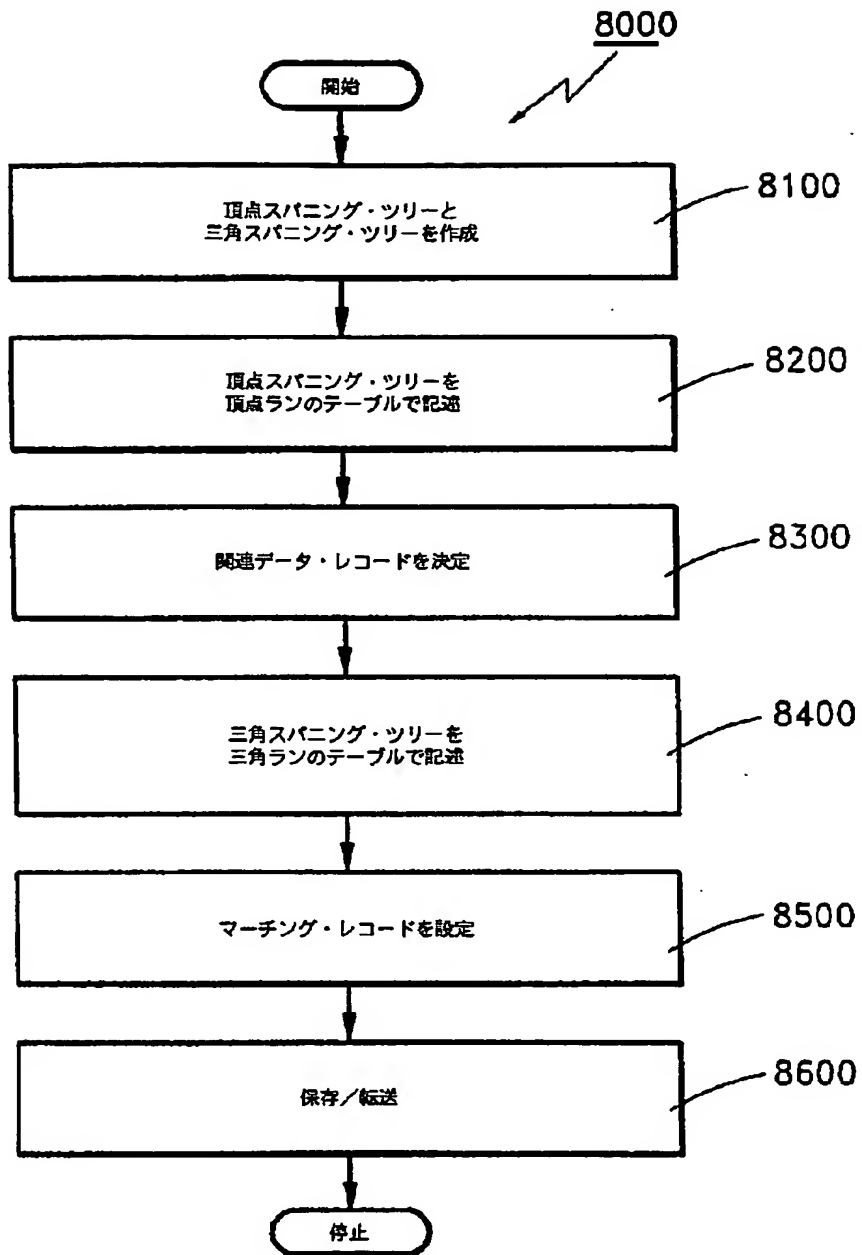
【図22】



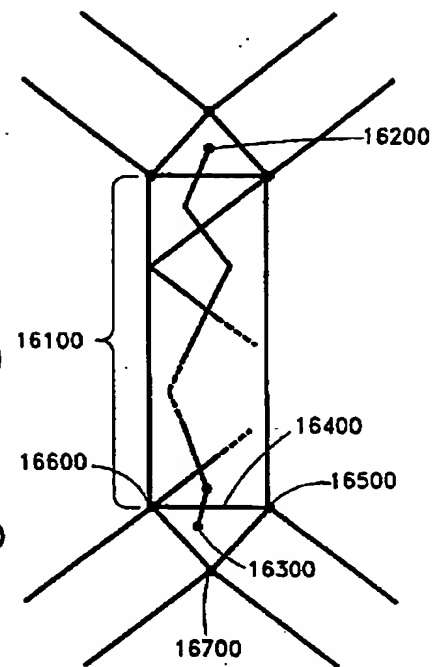
【図25】



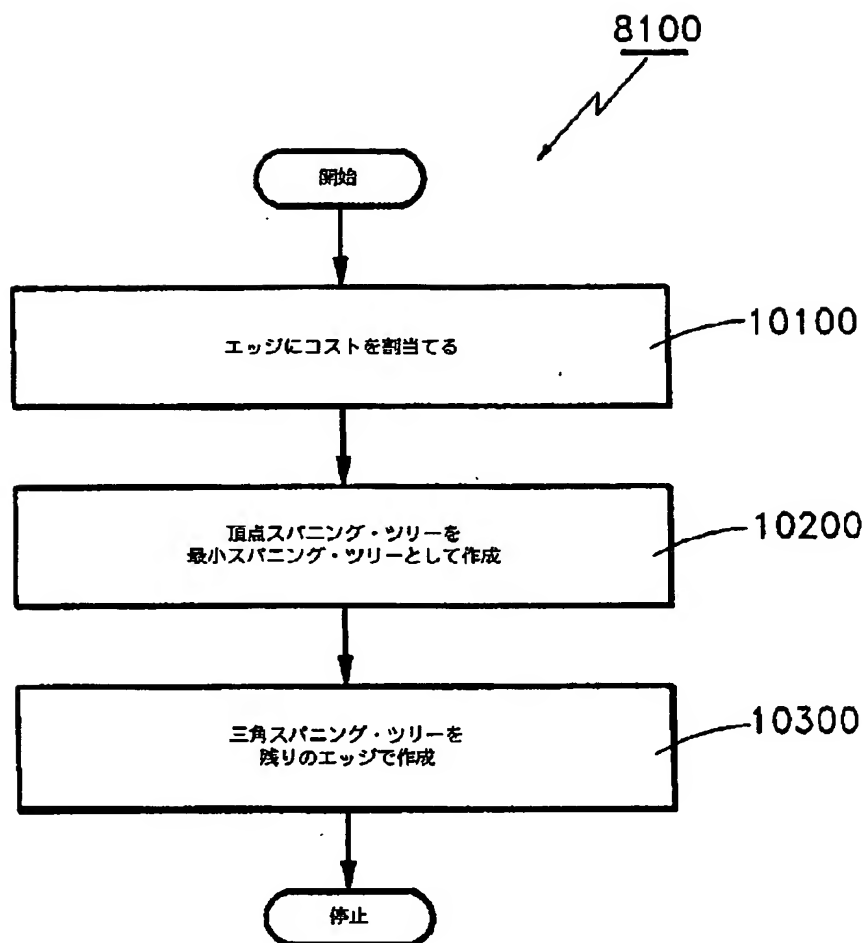
【図18】



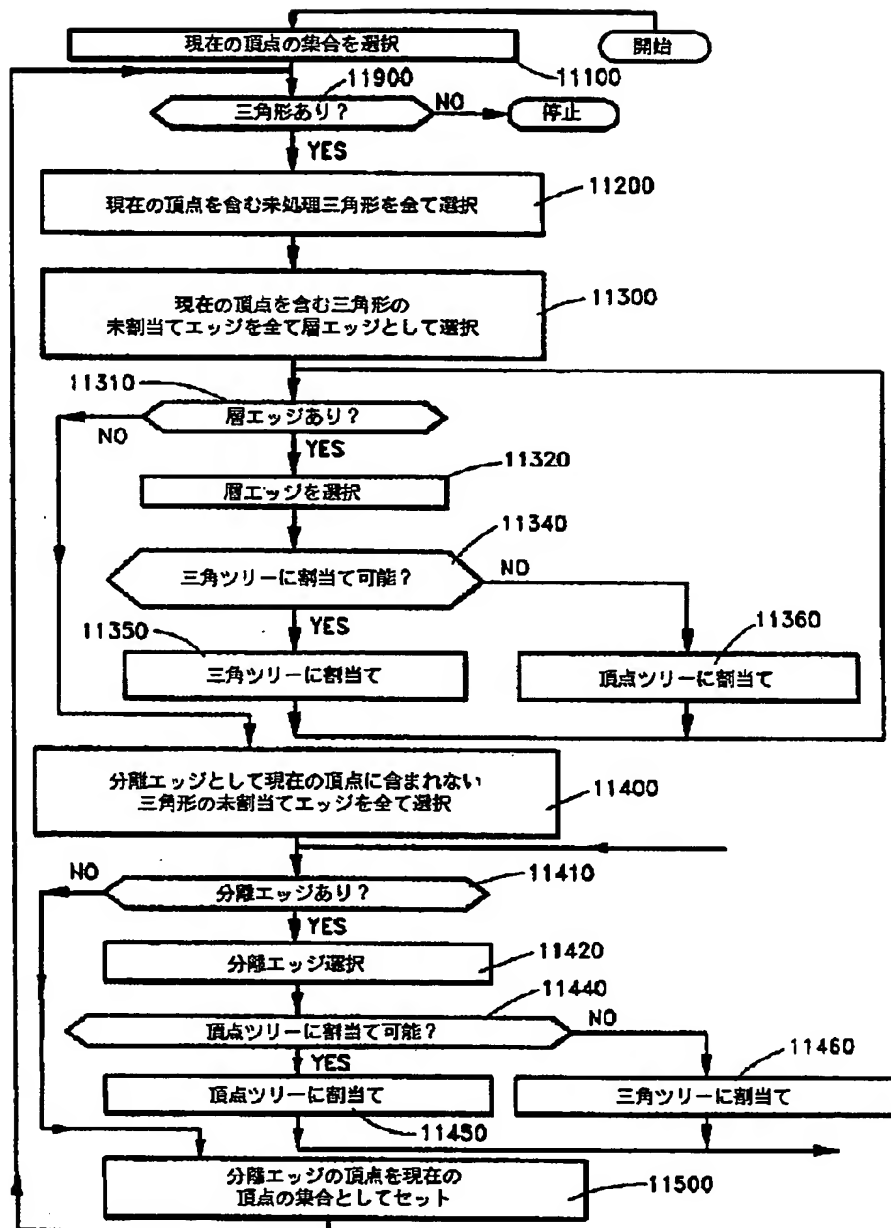
【図29】



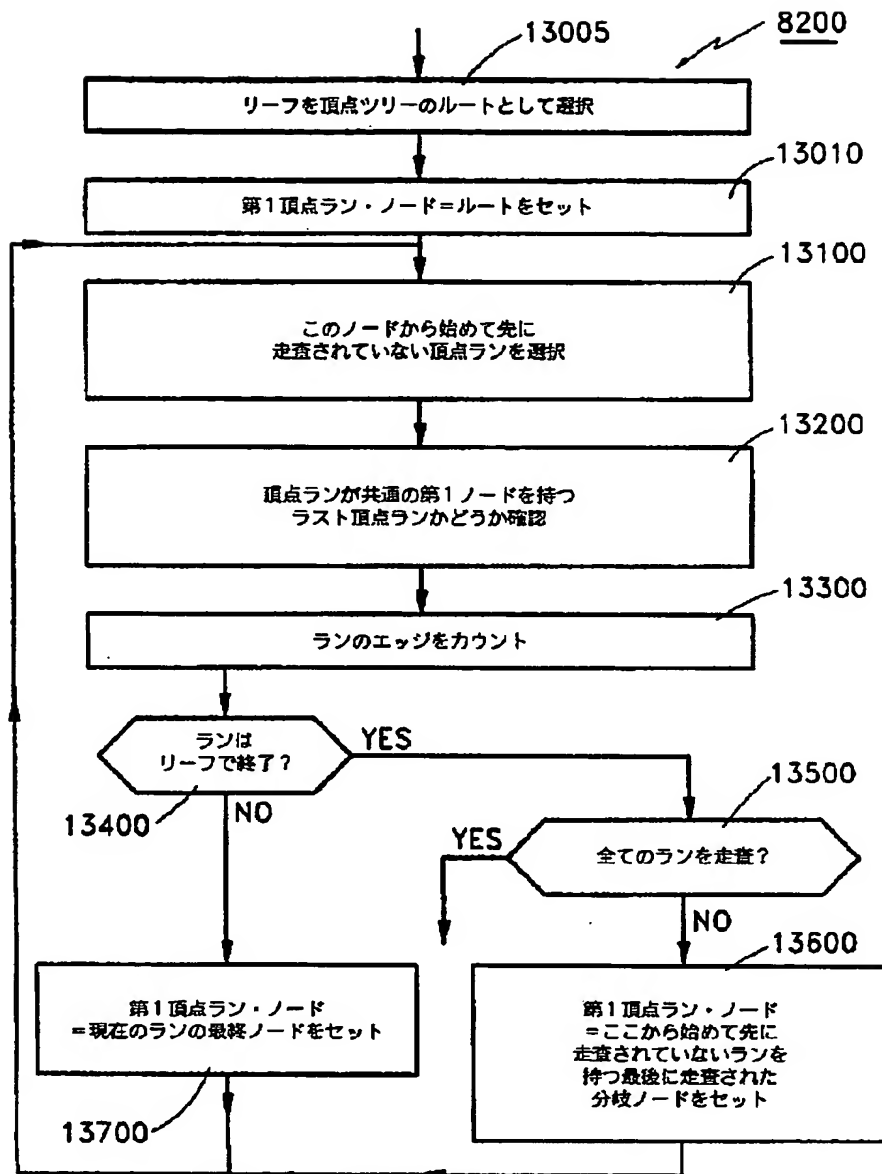
【図23】



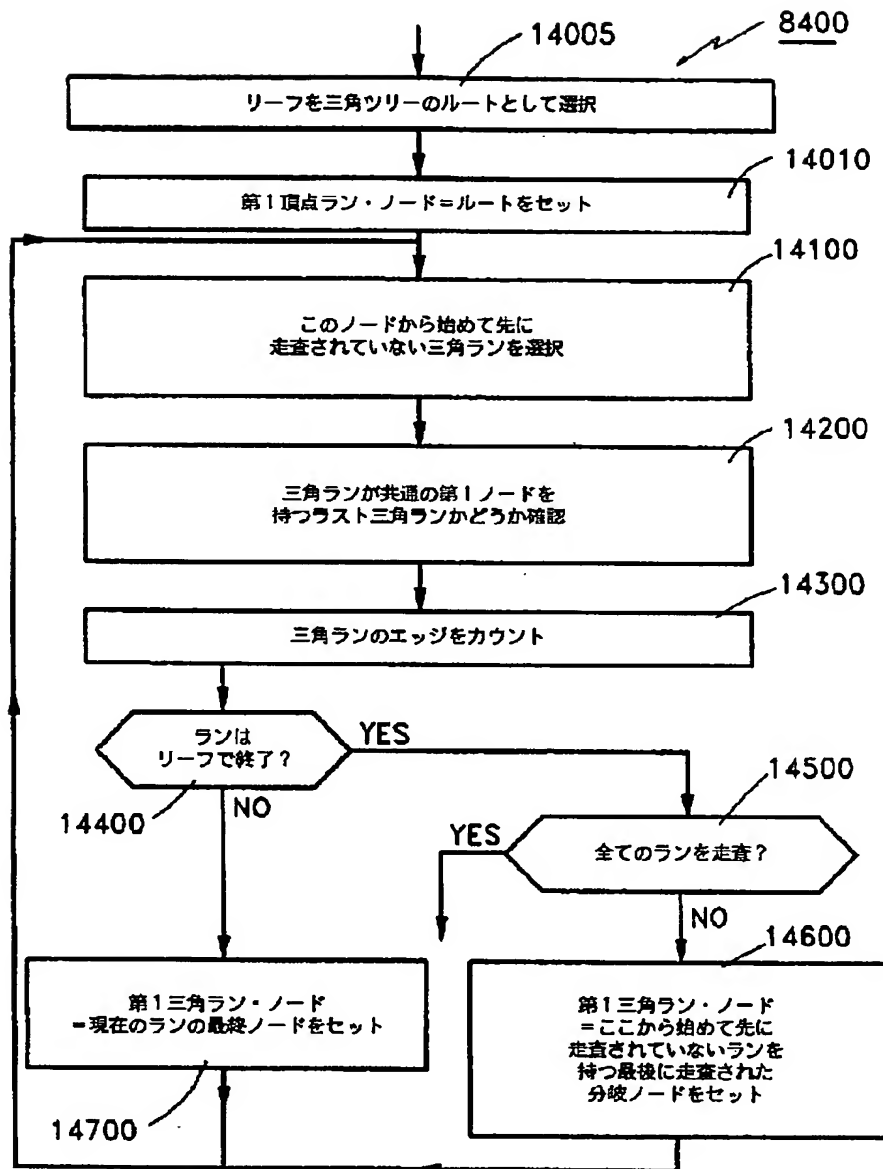
【図24】



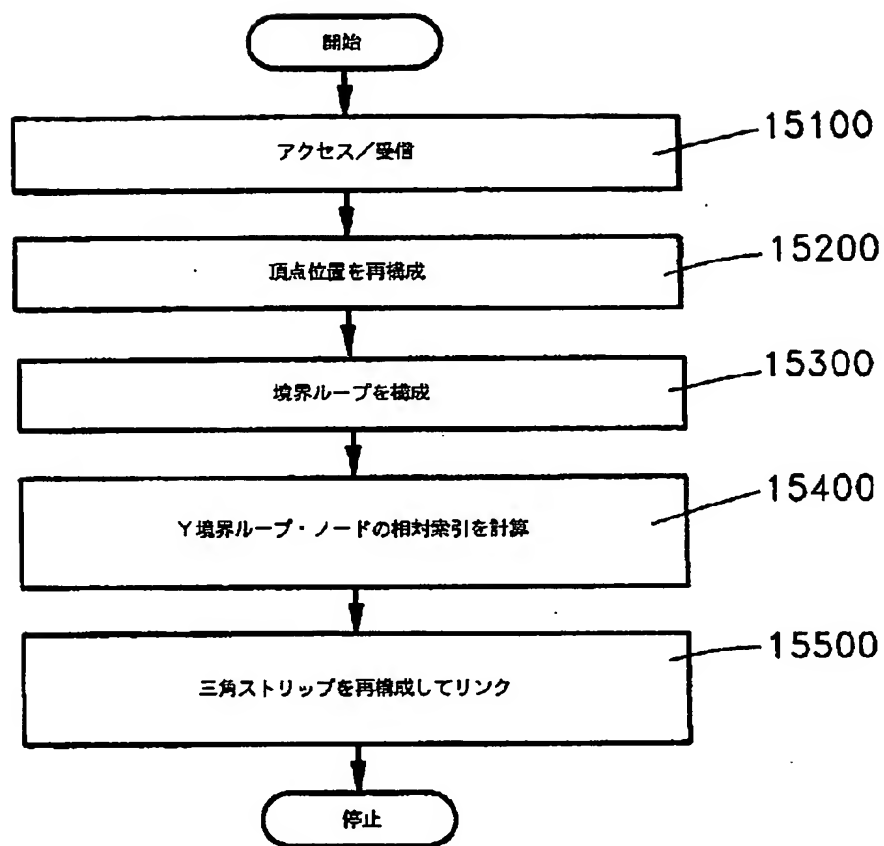
【図26】



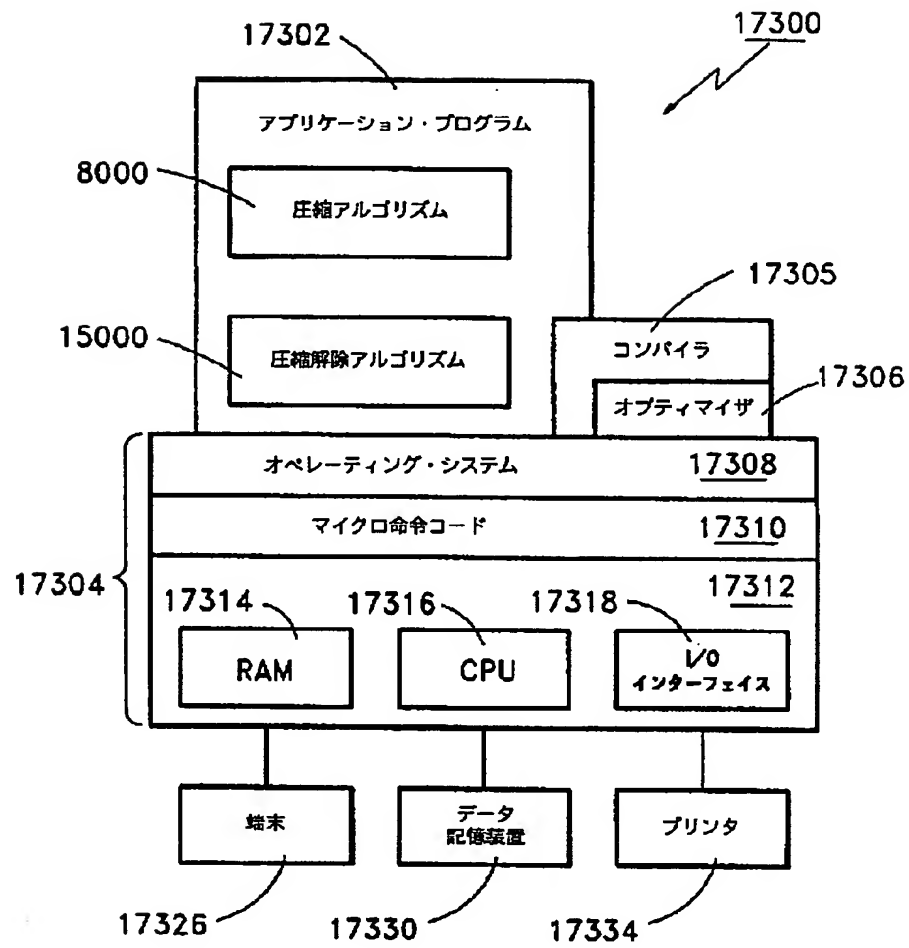
【図27】



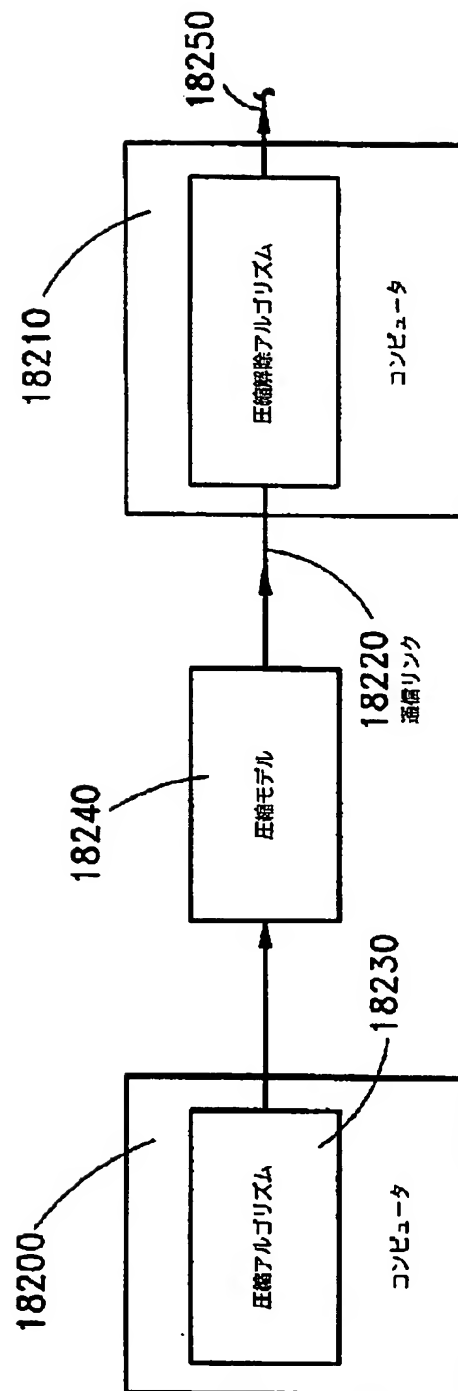
【図28】



【図30】



【図31】



フロントページの続き

(72)発明者 ジャレク・ジャロスロー・ロマン・ロシナ
ック
アメリカ合衆国10520、ニューヨーク州ク
ロトナーオンーハドソン、ジャスパー・ロ
ード 7

(72)発明者 ガブリエル・タウビン
アメリカ合衆国10530、ニューヨーク州ハ
ーツディル、ポー・ストリート 30